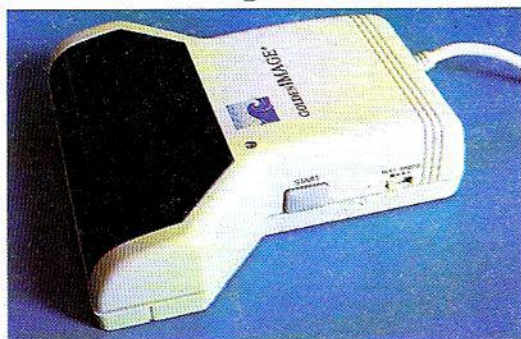
Anno XI - N. 83 - 25 Aprile 1991 - Sped. Abb. Post. Gr III/70 - CR - Distr.: Parrini

ACCESSORI AMIGA

Scanner e digitazione audio

AMIGA "C"

Gestire le memorie

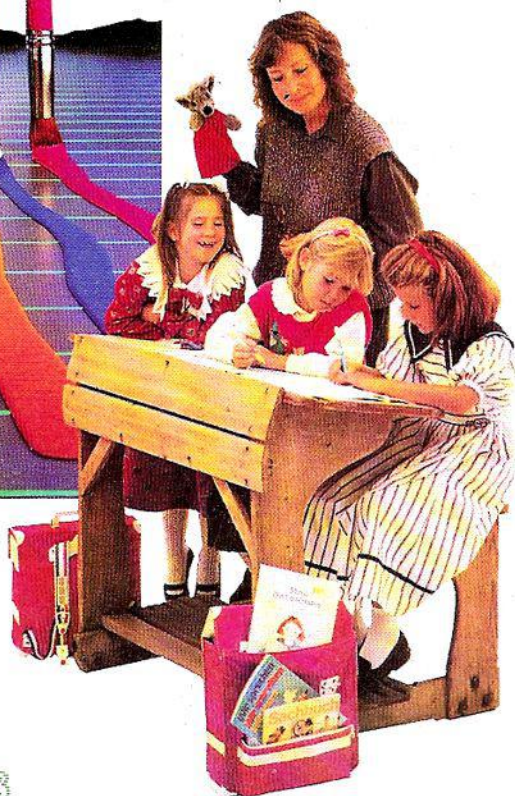
CONSIGLI

Come usare un word-processor

SCUOLA

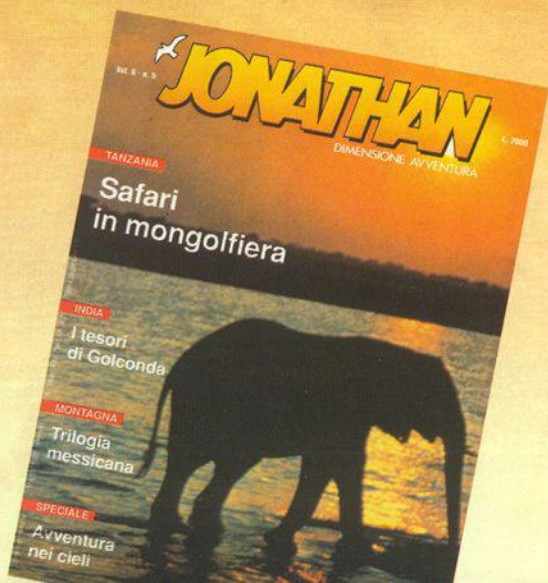
Un generatore di compiti in classe

DE LUXE PAINT

Un solo programma per due computer

GIOCARE AL LOTTO CON IL COMPUTER

Amiga facile • Divisione in sillabe • Sfida del mese • Grafica Amiga in 3-D • Amigazzetta 10 • Enciclopedia di routines "C" - Turbo Pascal - Quick Basic • Assembly 80X86 • La vostra posta • I comandi del Dos

Systems

JONATHAN
DIMENSIONE AVVENTURA
TANZANIA
Safari
in mongolfiera
INDIA
I tesori
di Golconda
MONTAGNA
Trilogia
messicana
SPECIALE
Avventura
nei cieli

Sommario

Amigames

Nessun videogame viene recensito questo mese; ciò per venire incontro alle numerose richieste dei nostri lettori, che preferiscono un maggior numero di pagine dedicate ad articoli tecnici.

Spazio 64

Amiga

Mondo Dos

Amiga + Ms - Dos

COMMODORE COMPUTER CLUB

Direttore: Alessandro de Simone
Coordinatore: Marco Miotti

Redazione / Collaboratori:
Davide Ardizzone - Claudio Baiocchi
Luigi Callegari - Umbero Colapicchioni
Donato De Luca - Carlo d'Ippolito
Valerio Ferri - Michele Maggi
Giancarlo Mariani - Domenico Pavone
Armando Sforzi - Dario Pistella
Fabio Sorgato- Valentino Spataro
Franco Rodella- Stefano Simonelli
Luca Viola

Direzione:
Via Mosè, 22 cap. 20090 OPERA (Mi)

Telefono 02 / 57.60.63.10
Fax 02 / 57.60.30.39
BBS 02 / 52.49.52.11

Pubblicità:
Leandro Nencioni (dir. vendite)
Via Mose', 22 20090 Opera (Mi)
tel. 02 / 57.60.63.10

Emilia Romagna:
Spazio E
P.zza Roosvelt, 4 cap. 40123 Bologna
Tel. 051 / 23.69.79

Toscana, Marche, Umbria
Mercurio s.r.l. Via Rodari, 9
S. G.nni Valdarno (Ar)
Tel. 055 / 94.74.44

Lazio, Campania
Spazio Nuovo
Via P. Foscari, 70
cap. 00139 Roma
tel. 06 / 81.09.679

Abbonamenti: Liliana Spina
Arretrati e s/w: Lucia Dominoni

Tariffe: Prezzo per copia L. 6000
Abbonamento annuo (11 fascicoli) L. 50000
Estero: L. 100000 - Indirizzare versamenti a:
Systems Editoriale Srl c/c 37952207 oppure
inviare comune assegno bancario non
trasferibile e barrato due volte a:
Systems Editoriale Srl (servizio arretrati)
Via Mose', 22
cap. 20090 OPERA (Mi)

Composizione: Systems Editoriale
La Litografica Srl Busto Arsizio (Va)

Registrazioni: Tribunale di Milano
n. 370 del 2/10/82

Direttore Responsabile: Michele Di Pisa

Spedizioni in abbonamento postale gruppo
III. Pubblicita' inferiore al 70%

Distributore: Parrini - Milano

Periodici Systems: Banca Oggi - Commodore Club (disco) - Commodore Computer Club - Commodore Computer Club (disco, produzione tedesca) - Computer - Computer disco - Electronic Mass Media Age - Energy Manager - Hospital Management - Jonathan - Nursing '90 - PC Programm (disco) - Personal Computer - Security - Software Club (cassetta ed. italiana) - TuttoGatto - Videoteca VR Videoregistrare

Editoriale

Chi sale e chi scende

Che l'andamento del mercato informatico fosse in fase di stasi, lo si sapeva già da tempo. Che il mercato dell'indotto (tra cui le riviste specializzate) dovesse seguire la stessa sorte, era nell'aria.

Una triste conferma deriva dall'annunciata chiusura, seguita dai fatti, di alcune testate del settore.

La colpa? Forse della distribuzione carente, del prezzo di copertina elevato, della concorrenza spietata, dell'eccessiva facilità di gestione del software disponibile (che rendeva superfluo il ricorso a dettagliati articoli di recensione). Colpa, comunque, non certo dei Redattori nè, tantomeno, dell'enorme passione che li spingeva a dedicare ore interminabili alla ricerca di procedure inedite, alla spiegazione di algoritmi forse complessi, alla descrizione accurata del funzionamento di un pacchetto applicativo.

La chiusura di una testata "concorrente" dovrebbe, secondo la spietata legge del mercato, essere motivo di soddisfazione (*"L'avevamo detto"* - forse dirà qualcuno); di trepide attese di aumento del numero di copie vendute (gli ex-lettori dovranno pur comprare altre riviste; perché non la nostra?); non certo di perplessità o di allarme.

Eppure tali notizie ci scuotono profondamente e ci rattristano forse quanto la delusione provata dai nostri colleghi, amareggiati non tanto dalla interruzione di introiti economici che la testata offriva loro (avranno maggiori gratificazioni in altre attività informatiche, ne siamo certi) quanto dalla delusione per aver visto diminuire, nel tempo, l'interesse degli appassionati verso il mondo informatico dilettevole, tanto impegnativo quanto in grado di garantire immense soddisfazioni.

E' il solito discorso della volpe e dell'uva? Gli acini dei grappoli possono essere paragonati ai lettori che, stanchi dell'impegno richiesto (e attratti dalla facilità dei videogames) hanno preferito desistere e abbandonare al suo destino la volpe che tenta di raggiungerli invano?

Comunque sia, per quanto riguarda la nostra testata, possiamo affermare che la svolta, molto sofferta, iniziata verso la fine del 1990, comincia a dare i suoi frutti. Il merito è, ovviamente, come sempre, dei nostri lettori (vecchi e nuovi) che sanno trovare soddisfazione in un hobby sempre più affascinante e misterioso.

Alessandro de Simone

LA VOSTRA POSTA

(a cura di A. de Simone)

Routines l.m. per C/64

Utilizzando un programma Assembler / Disassembler per C/64 mi piacerebbe registrare (dopo aver messo a punto e verificato il programma in codice mnemonico) il programma oggetto, in l.m. puro. Come posso fare?

(Matteo Bonfiglioli - Bologna)

Il manuale d'uso dell'assemblatore sicuramente riporta il modo di realizzare quanto richiedi (se l'hai letto con attenzione? A proposito: lo possiedi?...).

In ogni caso, tieni conto che quando si impartisce il comando di registrazione (da Basic o linguaggio macchina), il C/64 memorizza, su nastro o disco, il contenuto dell'area di memoria compreso tra due indirizzi.

Questi sono contenuti, nel solito codice Low / High, nelle due coppie di bytes 43/44 e 45/46. Naturalmente c'è bisogno, prima, di impostare il nome del file (187/188) e di altre cosette che, lo confesso, ho dimenticato da tempo dal momento che, in Redazione, il C/64 viene usato solo per provare il programma mensile che vedi pubblicato nella mini-rubrica "Spazio 64".

Non Commodore

Posseggo un computer di marca non Commodore. I listati che pubblicate, però, riesco a farli funzionare anche con il mio computer. Ciò vuol dire che il mio calcolatore è Commodore compatibile?

(anonimo principiante)

Evviva l'ingenuità! Il problema è diametralmente opposto a quanto prospettato. I computer Commodore sono Ms - Dos compatibili, e **non** viceversa. Il tuo computer, pertanto, è adattissimo per far girare i programmi e tutto ciò che vedi pubblicato sulle nostre pagine (limitatamente, è ovvio, al mondo Ms - Dos). Guai se alcuni programmi girassero solo sui Commodore: significherebbe che questi computer non sono compatibili al 100% con lo standard internazionale...

Da C/64 a Ms - Dos

Con il C/64 ed il word processor Easy Script ho scritto moltissimi testi (svolgo l'attività di insegnante). Mi piacerebbe utilizzare un computer Ms - Dos, ma l'incompatibilità dei dischetti me lo impedisce. C'è un sistema per convertire i files?

(Eduardo Aschettino - Chieti)

E' necessario collegare, tra loro, il C/64 ed il computer Ms - Dos tramite cavo seriale. I programmi di comunicazione (versione C/64 e Gw - Basic Ms- Dos) sono stati pubblicati, insieme al progettino hardware dell'interfaccia, sul **n. 49** di C.C.C. da richiedere presso il nostro servizio arretrati.

Led cieco

Posseggo un PC 10 III, dotato di due drive. Il led del secondo drive, però, non si illumina nè all'accensione dell'apparecchio nè duran-

Mini - risposte per mini - "concorso"

Nell'articolo sui comandi Ms - Dos (pubblicato sul n. 81) era riportata una mini - sfida che invitava i lettori a trovare una soluzione ad un problema apparentemente irrisolvibile.

Se, infatti, in ambiente Ms - Dos assegnate ad un file Batch il nome di un comando interno del Dos (come, ad esempio, `Date.bat` oppure `Ver.bat`) quando ne digitate il nome, per attivarlo, viene sempre "lanciato" il comando del Dos (**Date** oppure **Ver** nel caso specifico), procedura che impedisce l'intercettazione del nome del file batch.

Numerosissime sono state le risposte, praticamente tutte eguali tra loro (del resto, non si può inventare troppo...)

Due lettori, però, hanno per primi fornito la risposta al problema.

"Per ottenere che venga eseguito un programma dotato dello stesso nome di un comando del Dos *-scrive Davide Gnola da Ravenna-* occorre digitare...

```
Command /C c:\date.bat
```

...naturalmente facendo attenzione ai due tipi di barra inclinata e digitando i path corretti (nell'esempio riportato il file **date.bat** è presente nella dir principale del drive **C**)."

"Per attivare un file dotato di nome identico ad un comando del Dos *-scrive invece Andrea Preziosi di Sessa Aurunca-* è necessario invocare Debug, caricare e lanciare il programma con il comando G (Go)".

E' ovvio che quest'ultima procedura è più complessa (e, se mal condotta, rischia di inchiodare il computer), ma ha il vantaggio di essere molto versatile se, ad esempio, il programma dotato di nome eguale è di tipo Hidden (file nascosto) e si desidera inserire la procedura lavorando in linguaggio macchina.

Un terzo lettore propone un metodo brutale: "Impartendo Recover *-afferma Andrea Nardon di Trento-* tutti i nomi dei files verranno convertiti in File.000, File.001 eccetera". Bè, non era proprio ciò che cercavamo (bisognava attivare il file *senza* cambiarlo in alcun modo) ma apprezziamo l'ingegnoso artifizio.

Ai tre lettori verrà inviato, come promesso a suo tempo, un pacco dono contenente varie pubblicazioni della Systems editoriale.

te le operazioni con il drive stesso; operazioni che, tuttavia, si svolgono regolarmente.
(Andrea Fanella)

I led sono mini "lampadine", dotate di polarità, che si accendono solo se la corrente continua le attraversa nel verso giusto.

Di solito le due estremità dei led sono connesse con uno spinotto che, nella maggior parte dei casi, ha una forma tale da impedire un errato collegamento con la piastra madre del computer.

I casi, quindi, sono due: il led è fulminato e deve essere sostituito (spesa media, se affidi l'operazione ad un amico che se ne intende, di poche centinaia di lire); oppure, ancora più semplicemente, bisogna invertire il connettore, evidentemente mal inserito o non inserito affatto in fase di montaggio in fabbrica.

Schede per C/128

Che cartuccia mi consigliate per il mio C/128 che ho intenzione di usare in modo serio?
(Fabio B.)

La scheda (dal nome **"Computer Ms - Dos compatibile"**) esiste ed è davvero fantastica dal momento che non è necessario inserirla nel C/128, ma funziona anche "a distanza". Se, poi, il C/128 viene posto a distanza di sicurezza (ideale è la cantina o la soffitta) la scheda miracolosa è in grado di darti, in modo del tutto automatico e trasparente, soddisfazioni praticamente esagerate.

Le Poke di Ms - Dos

Sul C/64, computer che ho abbandonato a favore di un Ms - Dos compatibile, ero abituato ad impartire alcune Poke per modificare varie procedure (puntatori, vetto-

Orrore e sdegno

Alcuni lettori (tra cui **Marco Tonti, di Viserba**) lamentano la scarsa professionalità dimostrata da alcuni rivenditori di s/w (specie se il servizio viene offerto per corrispondenza) i quali vendono programmi, spesso in copia, malfunzionanti o pacchetti incompleti.

Le istruzioni allegate in fotocopia, poi, la dicono lunga sull'origine del s/w distribuito da tali commercianti.

Un invito alla prudenza, prima di acquistare numerosi titoli, è d'obbligo; il suggerimento di rivolgersi a rivenditori di s/w originale diventa una necessità sempre più inevitabile.

Amiga curiosità

E' stato annunciato un dispositivo che, in effetti, comprende un Amiga ed un lettore di CD in grado di "interpretare" software, musica hi-fi e immagini memorizzate su normali CD. L'apparecchio, che è privo di tastiera e comandabile tramite telecomando, dovrebbe inserirsi tra le apparecchiature domestiche di tipo educazionale (pessimo termine, ma di moda) e diventare familiare (almeno nelle intenzioni dei costruttori) quanto un videoregistratore. Per ora il s/w specifico (su dischi CD) è limitato a titoli come "Il Guiness dei primati" o a videogames educativi della LucasFilm e Walt Disney Co; è stato presentato nella scorsa edizione de "The Consumer Electronics Show".

Un'opinione

Egregi signori, vorrei esprimere la mia opinione sulla svolta editoriale della vostra Testata.

Compro la rivista ormai da tre anni ed è l'unica della quale, da allora, non ho perso più neanche un numero.

Scrivo per esprimere la mia opinione sulla svolta editoriale che è iniziata a Gennaio 1991. La riduzione di spazio dedicato ai vecchi C/64-128 è una buona scelta. Anche io ho posseduto un **C/128**, ma ora questi sistemi sono tecnologicamente sorpassati: per la risoluzione grafica, per la velocità di elaborazione, per l'incompatibilità del software, impossibilità di espansione ecc...

Vi ho scritto perché, incominciando a trattare parallelamente i sistemi **Ms - Dos** e **Amiga**, molti vi chiedono quale sia il sistema migliore. Credo di essere in una posizione privilegiata per rispondere dato che uso sia un PC IBM-AT, sia un Amiga e non sono redattore di una rivista.

Dal punto di vista hardware l'Amiga è più avanzato, dato che monta processori ad alta tecnologia (come Agnus), che gli Ms - Dos non hanno: la IBM ha sempre voluto mantenere la compatibilità con i vecchi sistemi XT, anche a costo di "restare indietro".

Dal punto di vista Software è invece il contrario, poiché gli Ms - Dos esistono dal 1980 mentre gli Amiga solo dal 1987 (l'Amiga 1000 fu un incidente...). Sapendo questo, la Commodore ha puntato su un utilizzo alternativo dei suoi computer: quello della videografica professionale / amatoriale e della musica con i vari Genlock.

Per questo gli Amiga non hanno drive velocissimi, nè software per tenere contabilità, fatturazione, o prodotti come il DB III (forse in futuro...).

Tuttavia, appassionato di elettronica quale sono, puntai sull'Amiga 500, nonostante ai tempi in cui dovevo decidermi per l'acquisto non desse alcuna garanzia.

Ebbene, non solo questo computer ha avuto un successo strepitoso, ma ultimamente sono apparse le schede di compatibilità Ms - Dos (addirittura AT) che testimoniano come un computer tecnologicamente avanzato possa emulare altri computer meno complessi.

Devo dire, comunque, che se avessi dovuto scegliere un computer per dirigere un'azienda, avrei preferito un più costoso Ms - Dos AT (gli XT sono sorpassati) grazie alla maggiore disponibilità di software specifico.

Veniamo ai linguaggi: fatemi il favore di non portare avanti un ennesimo corso sul linguaggio C. Un mio amico cervellone ha dedicato un anno e 600.000 lire per procurarsi compilatori e manuali (in inglese) nonché numerose riviste. Dopo tutto ciò è riuscito (pensate) a ricolorare in verde le palline del programma BOING.C.

Quando, con la vecchia versione (1.1) del mio **Ac-basic Compiler**, mi presentai a lui con un mio programma che cambiava colore alle palline ogni volta che toccavano terra, diede tutto alle fiamme, compilatore C, manuali ecc. Trovo invece brillante l'idea di trattare del portentoso **QuickBasic** e spero che parliate anche dei Basic superveloci di Amiga (AMOS e GFA).

Il Turbo Pascal è una buona idea, ma è un po' difficile.

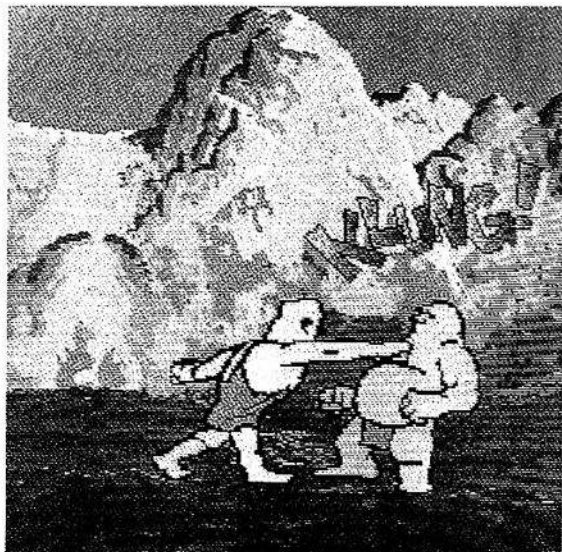
Trovo che la nuova impostazione sia valida, ma vi invito a tener sempre presente la "statura" del vostro lettore medio e a non esagerare con la complessità degli argomenti.

Ugo Spezza

ri e così via). C'è qualcosa di analogo in ambiente Ms - Dos senza ricorrere agli assemblatori (che non posseggo) ma limitandosi ad usare Pascal e Basic?
(Andrea Nardon - Trento)

E' una parola! Anzitutto, come ti sarai accorto, un sistema professionale come l'Ms - Dos è leggermente più complesso di un C/64.

Da un po' di tempo, pertanto, abbiamo iniziato diverse rubriche di un certo interesse che, impiegando linguaggi di alto livello (tra cui, "purtroppo", il linguaggio macchina) consentiranno di fare tutto ciò che era possibile sviluppare con il C/64.

Ed anche qualcosina di più...

Bit Movie '91

La quarta edizione della mostra si terrà nei locali del Palazzo del Turismo di Riccione dal **25 a 28 aprile 1991**. Ampio spazio sarà dato alla sezione musicale, affiancata da quella video, laboratorio, mostre fotografiche, sistema multivisione e, soprattutto, Amiga.

Tutte le sezioni, inutile dirlo, hanno come interprete principale software sofisticato, in grado di realizzare quanto di meglio esiste nel campo della musica e della grafica.

SYSTEMS EDITORIALE PER TE

La voce
Aggiunge al C/64 nuovi comandi Basic che consentono sia di far parlare il computer, sia di farlo Cantare! Diversi esempi allegati.

Cassetta: L. 12000 - Disco: L. 15000

Raffaello
Un programma completo per disegnare, a colori, con il C/64: linee, cerchi, quadrati, eccetera. Valido sia per disegno a mano libera che geometrico.

Cassetta: L. 10000

Oroscopo
Devi solo digitare la data di nascita e le coordinate geografiche del luogo che ti ha dato i natali. Vengono quindi elaborate le varie informazioni (case, influenze dei segni astrali, eccetera) e visualizzato un profilo del tuo carattere. Valido per qualsiasi anno, è indicato sia agli esperti sia ai meno introdotti. E' allegata una tabella delle coordinate delle più note città italiane e l'elenco delle ore legali in Italia dal 1916 al 1978.

Cassetta: L. 12000 - Disco: L. 12000

Computer Music
Cassetta contenente numerosi brani di successo da far eseguire, in interrupt, al tuo C/64 sfruttando, fino in fondo, il suo generatore sonoro (SID).

Cassetta: L. 12000

Gestione Familiare
Il più noto ed economico programma per controllare le spese e i guadagni di una famiglia.

Cassetta: L. 10000 - Disco: L. 10000

Banca Dati
Il più noto ed economico programma per gestire dati di qualsiasi natura.

Cassetta: L. 10000 - Disco: L. 10000

Matematica finanziaria
Un programma completo per la soluzione dei più frequenti problemi del settore.

Cassetta: L. 10000 - Disco: L. 20000

Analisi di bilancio
Uno strumento efficace per determinare con precisione i calcoli necessari ad un corretto bilancio.

Cassetta: L. 10000 - Disco: L. 20000

Corso di Basic
Confezione contenente quattro cassette per imparare velocemente le caratteristiche delle istruzioni Basic del C/64 e i rudimenti di programmazione. Interattivo.

Cassetta: L. 19000

Corso di Assembler
Un corso completo su cassetta per chi ha deciso di abbandonare il Basic del C/64 per addentrarsi nello studio delle potenzialità del microprocessore 6502. Interattivo.

Cassetta: L. 10000

Logo Systems
Il linguaggio più facile ed intuitivo esistente nel campo dell'informatica; ideale per far avvicinare i bambini al calcolatore.
Diversi esempi allegati.

Cassetta: L. 6500

Compilatore Grafico Matematico
Uno straordinario programma compilatore, di uso semplicissimo, che permette di tracciare, sul C/64, grafici matematici Hi-Res ad altissima velocità. Esempi d'uso allegati.

Cassetta: L. 8000

Emulatore Ms-Dos e Gw-Basic
Un prodotto, unico nel suo genere, che permette di usare, sul C/64 dotato di drive, la sintassi tipica del più diffuso sistema operativo del mondo. Ideale per studenti.

Solo su disco: L. 20000

Emulatore Turbo Pascal 64
Permette di usare le più importanti forme sintattiche del linguaggio Turbo Pascal (anche grafiche!) usando un semplice C/64 dotato di drive.
Ideale per studenti.

Disco: L. 19000

Speciale drive
Questo speciale fascicolo costituisce una guida di riferimento per le unità a disco del C64/128.
Comprende anche un velocissimo turbo-disk più la mappa completa della memoria del drive.

Fascicolo + disco: L. 12000

Utility 1
Un dischetto pieno zeppo di programmi speciali per chi opera frequentemente con il drive.

Disco: L. 12000

Utility 2
Seconda raccolta di utility indispensabili per realizzare sofisticate procedure di programmazione.

Disco: L. 15000

Graphic Expander 128
Per usare il C/128 (in modo 128 e su 80 colonne) in modo grafico Hi-res. Aggiunge nuove, potenti istruzioni Basic per disegnare in Hi-Res con la massima velocità in modalità 80 colonne.

Disco: L. 27000

Directory
Come è noto, a partire dal N. 10 di "Software Club" (la rivista su disco per l'utente dei "piccoli" computer Commodore), vengono riportati tutti i listati, in formato C/64-C/128, pubblicati su "Commodore Computer Club".
In precedenza tali listati venivano inseriti, mensilmente, in un dischetto, di nome "Directory", che oltre ai programmi di C.C.C. ospitava decine di altri file tra cui musiche nell'interrupt, giochi, listati inviati dai lettori e altro.
Ogni disco, dal prezzo irrisorio, contiene quindi una vera miniera di software. Ordinando i dischetti di "Directory" si tenga conto che al N. 1 corrispondeva il contenuto del N. 34 di "Commodore Computer Club", al N. 2 il N. 35 e così via.

Ogni dischetto: L. 10000

Super Tot '64
La nuova e completa edizione del programma Tot 13 con tutti i sistemi di riduzione e di condizionamento.
Ampia sezione dedicata alla teoria.

fascicolo + disco: L. 15000

Amiga Totospeed
Finalmente anche per Amiga un programma orientato alla compilazione delle schedine totocalcio.
Fai tredici conil tuo Amiga.

disco: L. 20000

SYSTEMS EDITORIALE PER TE

di Dario Pistella

DUE LISTATI PER RAGIONARE

Un paio di listatini in Basic, uno più breve dell'altro, possono farvi passare qualche ora piacevole.

Questo mese lo spazio dedicato al C/64 offre due algoritmi che, pur se presentati in modo piuttosto "grezzo", stimoleranno i principianti (e non) a rintracciare procedure più efficienti per raggiungere determinati scopi.

Si tratta dell'implementazione di un gioco di enigmistica e della ricerca del numero pigreco (3.1415...).

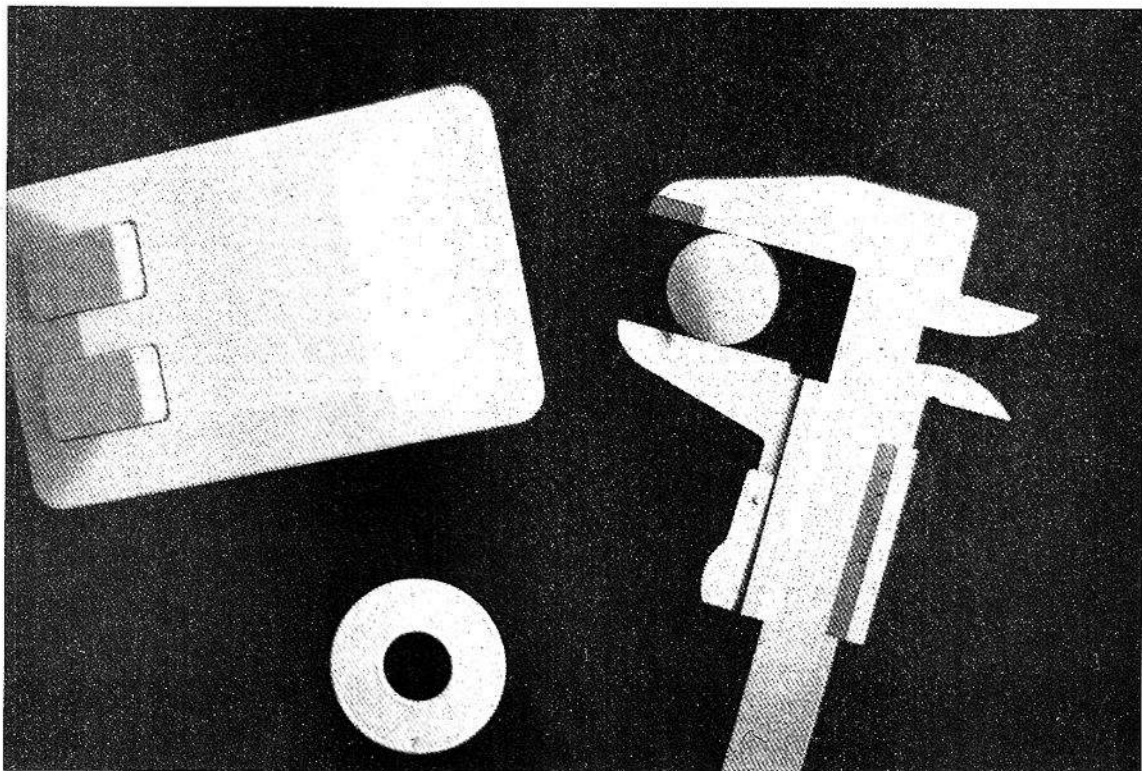
Il gioco

Spesso, sulla notissima **Settimana Enigmistica**, compare un gioco, denominato *"Sottrazione illustrata"*, che ha ispirato il programma di queste pagine.

Su più righe successive compaiono tanti caratteri, apparentemente privi di senso. Da ciascuna riga è necessario eliminare i caratteri appartenenti al nome delle immagini pubblicate sul settimanale (tante immagini quante sono le righe); i caratteri rimanenti, a soluzione avvenuta, costituiscono una frase celebre o un proverbio. Ad esempio, in...

rMosASsCHodEiRsAera
CAbelVteAmLpLOosispera

...il proverbio *"rosso di sera bel tempo si spera"* è semioccultato dai vocaboli MASCHERA (primo rigo) e CAVALLO (se-

cond rigo). Per meglio chiarire il concetto, i due nomi sono stati riportati in maiuscolo, ma è chiaro che, in realtà, i caratteri compaiono tutti in maiuscolo oppure in minuscolo; altrimenti la soluzione sarebbe troppo semplice!

Il programma

Il breve listato non fa altro che leggere la frase da indovinare (nient'altro che una stringa associata a DATA di riga 430) e , dopo averla "smembrata", la fa apparire, in alto sullo schermo, frammischiata ad altri caratteri casuali. Naturalmente il gruppo dei caratteri della frase da indovinare è anch'esso casuale; ciò per evitare che i caratteri stessi si susseguano con prevedibile successione.

Per giocare è necessario essere in due: mentre il primo, modificando la riga 430, inserisce la frase da indovinare, l'altro non deve assistere alla modifica. Dopo il Run, spostandosi "lungo" la frase visualizzata (mediante i tasti cursore), bisognerà premere la barra spaziatrice per occultare il carattere su cui si è posizionati; in caso di errore, posizionando il cursore sulla cella "vuota", sarà possibile individuare (in colore diverso) il carattere spento per errore, e farlo riapparire premendo nuovamente la barra.

Una volta individuata la frase, la pressione del tasto Return interromperà il gioco e verrà visualizzato il numero di caratteri minimo che era necessario occultare per far apparire la frase. L'ideale, ovviamente, è rappresentato dal numero giusto.

Chi, però, indovina la frase prima di aver occultato tutti i caratteri necessari, conseguirà un punteggio maggiore. Si noti che, nel determinare il numero dei caratteri occultati, il programma conteggia anche i caratteri che sono stati fatti apparire e scomparire più di una volta.

```
100 rem sottrazione di caratteri
110 :
120 poke53281,0:printchr$(147);:c=54272
130 x=int(rnd(-1)*5):rem seme
140 read a$:l=len(a$):wa=0:wb=0:t=1: for i=1 to l
150 x=int(rnd(1)*5)+1:y=65+int((rnd(1)*26))
160 print mid$(a$,t,x);chr$(y);:wa=wa+1:t=t+x
170 if t<=l then next
180 x=1024:g=1:dim m%(255):fori=0 to 254:m%(i)=1:next
190 get x$: ifx$=""then gosub 370:goto 190
200 ifx$="■"then gosub270:goto190
210 ifx$="■"then gosub290:goto190
220 ifx$="■"then gosub310:goto190
230 ifx$="■"then gosub330:goto190
240 ifx$=chr$(32)then gosub350:wb=wb+1:goto190
250 ifx$=chr$(13)then fori=1to10:print:next:printwb,"(erano"wa")":end
260 goto190
270 g=g+1:ifg>6 theng=6:return
280 x=x+40:return
290 g=g-1 :ifg<1 theng=1:return
300 x=x-40:return
310 s=s+1 :ifs>39 thens=39:return
320 x=x+1:return
330 s=s-1 :ifs<0  thens=0 :return
340 x=x-1 :return
350 if m%(x-1024) >0 then poke c+x,0:m%(x-1024)=0:return
360 poke c+x,1:m%(x-1024)=1:. return
370 h=m%(x-1024):u=peek(x)
380 if h>0then pokex,u+128:gosub410:pokex,u:gosub410:return
390 if u<>32 then pokec+x,5:gosub410:pokec+x,0:gosub410:return
400 pokex,32+128:pokec+x,5:gosub410:pokec+x,0:pokex,32:gosub410:return
410 forj=1to25:next:return
420 :
430 data commodorecomputerclube'unarivistadellasystemseditoriale
440 end

ready.
```

Migliorie

Come si può notare, il listato, decisamente rozzo, inserisce caratteri casuali all'interno della frase da individuare. L'ideale sarebbe di "mescolare" tra loro due frasi di senso compiuto, e indovinarle entrambe.

Un'altra modifica potrebbe essere quella di inserire un Input che consenta, al primo giocatore, di digitare le frasi che l'altro deve indovinare, senza ricorrere alla modifica della riga 430 (attenzione all'input controllato!).

Un'ultima modifica potrebbe essere quella relativa alla digitazione delle due frasi, nel caso si pervenga alla soluzione prima della fine, ed alla corrispondente determinazione del punteggio.

Pi greco

Sappiamo tutti che il numero pigreco vale **3.141592654...** e ce lo dimostra il visore di una qualsiasi calcolatrice tascabile scientifica che ne consente la visualizzazione.

Ma come si perviene a quel risultato e perché si afferma che non c'è limite al numero di cifre decimali presenti dopo la virgola?

Il famoso numero deriva dallo sviluppo di una serie matematica e il metodo per determinarlo non è unico.

Uno di questi è il seguente:

Pigreco = 4/1 - 4/3 + 4/5 - 4/7 +4/9 - 4/11 + 4/13...

In pratica, pigreco deriva da una successione di somme algebriche di frazioni; queste presentano, tutte, 4 al numeratore mentre, al denominatore, figura la successione dei numeri dispari (1, 3, 5 eccetera). Il segno, infine, cambia alternativamente dą positivo a negativo.

Ecco, dunque, perché il numero di cifre decimali è infinito; questo è anche il motivo per cui non vale la pena proseguire oltre un certo numero di cifre: nei casi pratici, come sappiamo, l'approssimazione **3.14** è più che sufficiente.

Se, però, vogliamo sviluppare l'algoritmo, il secondo programma di queste pagine può esserci di aiuto.

Subito, comunque, sorge un problema: pur se è possibile impostare un ciclo infinito, il procedimento cozza, prima o poi, con la precisione interna della macchina.

La somma delle varie frazioni, infatti, non può non tener conto del limitato numero di cifre decimali elaborabile dal microprocessore (o meglio: dall'algoritmo matematico inserito nel linguaggio adoperato).

Il programma

A causa dei motivi prima esaminati, quindi, il programma provvede a visualizzare, frazione dopo frazione, il valore raggiunto dopo ciascuna elaborazione (indicata a sinistra).

Si può notare che alla prima, terza, quinta... elaborazione corrisponde un valore maggiore di pigreco; mentre alla seconda, quarta, sesta... elaborazione, come intuitivo, un valore minore.

A mano a mano che il conteggio aumenta, il valore determinato si approssima sempre più al valore classico di pigreco. Come si può osservare nelle Rem (righe 170 / 180), dopo circa **16000 cicli** il valore determinato è ancora abbastanza lontano da quello a noi noto, "incorporato" nelle Rom del C/64 (notare il costante confronto di riga 130). La mancata coincidenza deriva dalla (im)precisione interna del computer, dall'inevitabile lungaggine della procedura (tipica di ogni "serie") oppure, più banalmente, dall'errato algoritmo? A voi l'ardua sentenza.

In ogni caso, possiamo certamente affermare che chi ha proseguito i calcoli (alcune centinaia di anni fa) ed è pervenuto al valore che tutti noi oggi conosciamo (anche se limitatamente alle prime dieci cifre decimali) ha avuto un bel fegato, soprattutto se si tiene conto che, a quei tempi, i calcoli si eseguivano a mano libera...

```
 1  6.31839562  3.14159265
 3  2.31839561  3.14159265
 5  3.65172895  3.14159265
 7  2.85172895  3.14159265
 9  3.42315752  3.14159265
 11  2.97871308  3.14159265
 13  3.34234944  3.14159265
 15  3.03465713  3.14159265
 17  3.3013238  3.14159265
 19  3.06602968  3.14159265
 21  3.276556  3.14159265
 23  3.08607981  3.14159265
 25  3.25999285  3.14159265
 27  3.09999285  3.14159265
 29  3.248141  3.14159265
```

```
100 REM DETERMINAZIONE DI PIGRECO
110 :
120 X=1
130 IF PI=π THEN END
140 PRINT"■"X"■"PI;π
150 IF DV=0 THEN Y=4/X:PI=PI+Y:X=X+2:DV=1:GOTO130
160 Y=-4/X:PI=PI+Y:X=X+2:DV=0:GOTO130
170 REM 15955  3.14171804
180 REM 15957  3.14146733
190 REM π=4/1 -4/3 +4/5 -4/7 ECCETERA
200 END

READY.
```

di Alessandro de Simone

COME USARE UN WORD PROCESSOR

Un articolo da leggere attentamente per realizzare, senza difficoltà, testi e documenti

Alla categoria dei word processor appartengono i programmi professionali che permettono la battitura di testi, e documenti in generale, con possibilità di richiamo dei testi stessi, modifiche, aggiunte, manipolazioni varie, eccetera.

All'inizio dell'era informatica furono realizzati dei **Text Editor** che consentivano la redazione di testi, ma erano piuttosto limitati, pur se nettamente preferibili alla semplice macchina da scrivere.

In seguito, con lo sviluppo della tecnologia, e del software in particolare, si è giunti alla realizzazione di programmi strordinariamente ricchi e potenti, in grado di trasformare anche piccoli computer domestici (come il **C/64**, l'**Amiga** o i compatibili **Ms - Dos**) in vere stazioni di videoscrittura.

Non sempre, però, il libretto di istruzioni, quando c'è e quando è scritto in italiano, risulta chiaro, nè risponde con semplicità agli inevitabili **dubbi** di un principiante, nè, tantomeno, è in grado di spiegare il perchè di **apparenti malfunzionamenti**, legati, inutile dirlo, all'inesperienza dell'utilizzatore.

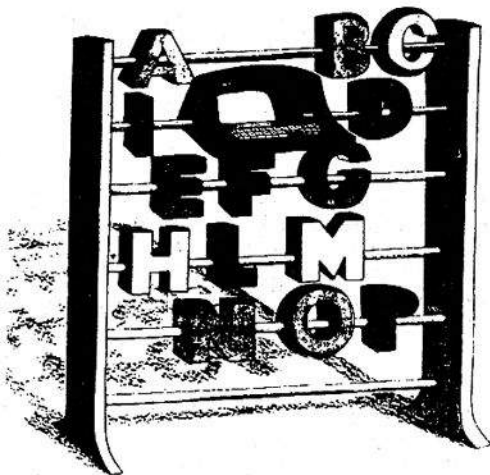
Cerchermo, pertanto, di chiarire alcuni concetti base comuni a qualsiasi word processor (d'ora in poi: **w/p**); ma ci riferiremo, in particolare, al notissimo **Word Star** (versione **5.5**) che, per quanto ci risulti, è il più semplice e diffuso w/p in ambiente Ms - Dos.

Nei panni del computer

Uno degli errori più comuni che l'utente "medio" commette, è certamente quello di considerare il calcolatore simile ad una persona umana. Pur se i ragionamenti da seguire portano agli stessi risultati, non dobbiamo dimenticare che **il computer "vede" un testo in modo diverso da come lo vediamo noi**.

L'uomo realizza documenti cartacei secondo uno schema di utilità personale, legato alla miglior combinazione possibile tra velocità di lettura e comprensione. Centinaia di anni di esperienza hanno convinto l'uomo che, nel caso di documenti scritti, la soluzione ideale è rappresentata dall'utilizzo di fogli di carta rettangolari, in cui il messaggio viene riversato, da sinistra a destra e dall'alto in basso, servendosi di caratteri alfanumerici posti l'uno di seguito all'altro (i cinesi hanno esperienze diverse; lo scopo, tuttavia, è lo stesso: chiarezza e velocità di comprensione).

In linea teorica sarebbe possibile scrivere, e leggere, un documento su un nastro sottile di carta (una bobina, insomma) lungo centinaia di metri. La scomodità di utilizzo, però, è incompatibile con le prerogative che si desiderano in un testo scritto.

Non dobbiamo pensare, tuttavia, che l'idea di riversare su un lungo nastro una

comunicazione debba esser scartata in assoluto: quando sosteniamo una conversazione orale, infatti, non facciamo altro che realizzare, di fatto, un lungo "nastro" in cui le parole si susseguono l'una dopo l'altra, senza interruzione, senza divisioni in sillabe, senza *"a capo"* che sono, invece, strumenti indispensabili in un documento scritto.

Con ciò vogliamo dire che, a seconda dei casi, si utilizza un sistema di comunicazione piuttosto che un altro; ma tutti, comunque, indistintamente finalizzati per l'ottimale scambio di informazioni.

La chiacchierata finora sostenuta è servita solo ad affermare che ognuno segue il sistema di comunicazione più adatto alla propria struttura. Il computer non ha momenti "alternativi" (come per noi potrebbero essere individuati nello *"scritto"* e nel *"parlato"*) ma ragiona sempre, e solo, basandosi su di una successione di byte, l'uno posto di seguito all'altro: tale struttura si riscontra nel modo stesso di ragionare (un'istruzione dopo l'altra), di inviare dati (sia in modo parallelo che seriale), di stampare messaggi su stampante (sia in modo testo che in modo grafico) sia nel disegnare in grafica.

E' solo nostro, quindi, il compito di raggruppare più dati tra loro e conferirvi un'immagine (o come si dice in gergo: **formattarli**) secondo uno schema più adattabile alla nostra struttura mentale.

Anche in un w/p, dunque, il computer si accontenterebbe di scrivere tutti i caratteri alfanumerici del nostro testo l'uno di seguito all'altro. Noi, però, per assegnare al risultato dell'elaborazione una forma più familiare, costringiamo il computer a riversare su video i dati in modo da simulare una pagina dattiloscritta; non potete non convenire che la struttura di rappresentazione di un qualsiasi monitor è stata realizzata, con non pochi sacrifici, per accontentare la nostra "pigrizia" o, se preferite, la difficoltà di adattamento ad altri sistemi di visualizzazione.

Concludiamo, quindi, sottolineando che ciò che vediamo sullo schermo (sia che scriviamo un programma o un testo) è un adattamento *"per pagine"* (= schermate) di una situazione che, per sua natura, potrebbe invece essere rappresentata da un'**unica riga**, lunga quanto l'intera memoria disponibile nel computer utilizzato.

Tali premesse saranno preziose tra breve, quando cercheremo di capire alcuni inspiegabili malfunzionamenti dei w/p.

Vari tipi di Word processor

Esiste più di una categoria di w/p. Possiamo, anzitutto, suddividerli in quelli che consentono l'eventuale divisione automatica, in sillabe, delle parole presenti alla destra di un rigo di testo (come Word Star).

L'altra categoria, naturalmente, non prevede questa comoda(?) possibilità: nel caso in cui, in un rigo di testo, l'ultima parola a destra non possa essere ospitata per intero sul rigo stesso, verrà riprodotta, **automaticamente**, come prima parola sul rigo successivo.

Quest'ultima categoria di w/p era la più diffusa ed il motivo è semplice: le regole grammaticali per la divisione in sillabe non sono le stesse per tutte le lingue; ne consegue che un w/p americano creava difficoltà per gli utenti italiani o tedeschi e, di fatto, la commercializzazione del programma ne avrebbe risentito. Oggi, invece, quasi tutti i w/p di un certo livello offrono l'opzione di divisione in sillabe, selezionando la lingua desiderata.

Qualunque W/p può essere classificato anche in base alla **simulazione video**: totale o parziale.

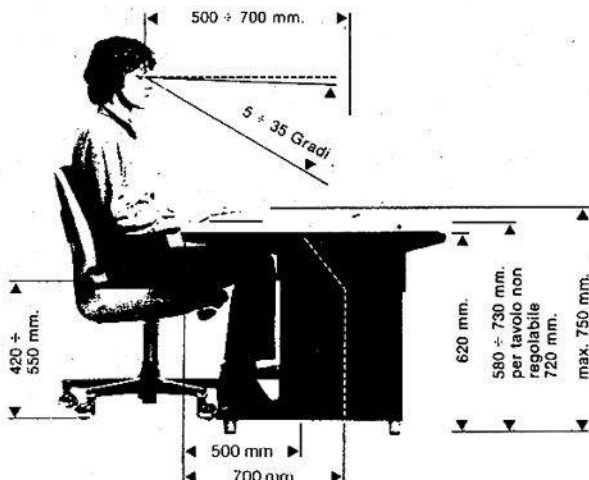
Con l'aggettivo "totale" intendiamo dire che, durante la digitazione di un testo, appare sullo schermo esattamente ciò che verrà riprodotto sulla stampante. Tale opportunità è presente, in modo ottimale, soltanto su computer professionali che, dotati di schermo verticale, simulano, per quanto possibile, il formato di un foglio di carta.

Altri computer meno costosi consentono, tuttavia, la possibilità **WYSIWIG** (What you see is what you get = ciò che vedi è ciò che ottieni, cioè stampi) ma è necessario qualche sacrificio: con gli Ms - Dos compatibili è possibile visualizzare, su schermo, solo **porzioni** di pagine, a causa del formato del video (di solito 80 righe x 25 colonne). Con le moderne schede grafiche (che consentono di visualizzare anche 125 colonne e 60 righe) il problema risulta parzialmente risolto.

Come fare, dunque, con i computer limitati ad un massimo di 80 x 25?

Il metodo più sbrigativo (e, quindi, più seguito) è quello della *finestra* (window) grazie al quale lo schermo viene trasformato in una sorta di buco della serratura (di ampiezza 40 x 25 oppure 80 x 25) che si sposta sul documento mostrandone una parte per volta. Con un po' di fantasia è quindi possibile immaginare l'aspetto finale del documento che verrà riversato su stampante.

Si va ad incominciare

Vediamo ora, finalmente, di trovare un riscontro con quanto abbiamo finora affermato, limitandoci alla descrizione di ciò che accade usando WS.

Appena si lancia WS, compare un menu di inizio contenente numerose opzioni, delle quali, per ora, interessano solo alcune.

Anzitutto, vi consigliamo di procurarvi un dischetto formattato (e vuoto) da inserire in uno dei drive a vostra disposizione (ci riferiremo, per semplictà, al drive **A:**). Quindi, selezionando **L** (cioè: opzione *selezione indice / drive*) digitate, appunto, **A** e premete Return. Alla ricomparsa del menu principale, premete **D** (apertura **D**ocumento) e digitate: **esempio1**; questo sarà considerato come *nome del file di testo* che intendete scrivere. Dal momento che, nel drive A, è presente un disco vuoto, WS chiederà *"Il file non esiste; devo crearlo?"* cui risponderete, ovviamente, **S** (se disponete della versione americana premerete, invece, **Y**).

Subito lo schermo verrà cancellato ad eccezione della prima riga. In questa, a sinistra, compaiono, tra gli altri, il messaggio A:ESEMPIO1 (che non necessita di commenti), **P1** (pagina 1, siamo all'inizio!); **R1**, **C1** (riga 1, colonna 1 in cui lampeggia il cursore); **Inser** (su cui torneremo dopo).

Prima di andare oltre, scrivete una frase qualsiasi (il vostro nome, ad esempio) e correggete eventuali errori di battitura con il tasto **BackSpace**.

Una volta digitata la frase (scrivetela abbastanza lunga e non curatevi di eventuali errori di battitura), spostatevi lungo lo schermo utilizzando i tasti cursore e se, andando in basso, ciò che avete scritto viene spostato in alto, fino a scomparire dalla vista, niente paura! Risalite con il cursore, oppure, più semplicemente, premete il tasto **Control** e, mentre lo tenete abbassato, il tasto **Home**: immediatamente il cursore lampeggerà sulla prima colonna della prima riga.

Prima di procedere è bene sottolineare che il tasto **Shift** (come pure il tasto **Control**) ha effetto solo se viene **tenuto** abbassato mentre si batte un altro tasto; in altre parole, con il termine **Control + Home** (e analoghi) intendiamo:

* premere uno dei due tasti Control e tenerlo abbassato.
* premere, con l'altra mano, il tasto Home.

Se non rispettate l'ordine di pressione dei tasti, oppure se il tasto Control non è tenuto premuto, non si verifica quanto desiderate. Con i comandi "doppi" (esempio **Control + K + V**) è sufficiente tener premuto Control solo la prima volta (cioè con K); il secondo tasto (V, nell'esempio specifico) potete premerlo anche da solo.

Bene, torniamo ai nostri esperimenti: premete Control + Home e, qualunque cosa abbiate scritto, o compaia in questo momento sul video, scrivete, di seguito, il vostro nome, cognome e indirizzo. Se prima c'era scritto qualcosa, (stiamo parlando del primo rigo) ciò che ora digitate si sovrappone ai caratteri già presenti se, sul primo rigo in alto, **non** compare la parola Inser.

Nel caso questa dovesse, appunto, apparire, premete il tasto **Insert** (oppure **Ins** o **Inser**, a seconda della tastiera posseduta) per "spegnerla". Approfittate per notare che, ad ogni pressione di un tasto alfanumerico, i valori associati a "**R**" e "**C**" (in alto a sinistra del video) vengono modificati automaticamente.

Ora faremo un esperimento: premete il tasto Inser: contemporaneamente la parola Inser, posta più o meno al centro della prima riga di schermo, scompare. Premete nuovamente Inser e facciamo qualche considerazione: il tasto Inser è uno di quei tasti cosiddetti "bistabili". Ciò significa che premuto una volta (o un numero di volte **dispari**) viene **attivato**; premuto una seconda volta (o un numero di volte **pari**) viene **annullato**.

In conclusione, per attivare Inser (vedremo tra breve a che serve, anche se l'avete già capito) è necessario premere il tasto, una sola volta, e rilasciarlo.

Supponiamo, ora, di aver digitato numerose parole (cercate di riempire una schermata, anche scrivendo a casaccio), posizionatevi in un punto qualsiasi del testo scritto (con i tasti cursore) e premete **Control + K**. Subito compare un menu diverso dal precedente.

Premendo **B**, il menu scompare ma, nel punto preciso in cui prima lampeggiava il cursore, compaiono tre caratteri colorati **<B>** che, per quanto tentiate, non risucirete a cancellare. Spostatevi, ora, di qualche carattere più a destra (o più in basso, ma non a sinistra, nè in alto!) e premete **Control + K + K**. Noterete che il testo compreso tra <B> ed il punto in cui era presente il cursore ha cambiato colore.

Avete definito un "**blocco**" di testo: ora ne potete fare ciò che volete, cioè:

Control + K + Y Il blocco viene cancellato.

Control + K + C Il blocco viene riprodotto a partire dal punto in cui si trova il cursore.

Control + K + V Il blocco viene cancellato dalla posizione precedente e trasportato a partire dal punto in cui è presente il cursore al momento dell'ordine.

Control + K + W Il blocco viene memorizzato su disco (bisogna, ovviamente, assegnare il nome).

Altri esperimenti potrete condurli da soli facendo prove sul menu che compare con **Control + K**. Vi consigliamo, prima, di definire un blocco nel modo indicato.

Se, per errore, avete premuto Control + K, sarà sufficiente premere il tasto **Esc** per annullare il comando stesso e tornare in modo **Edit** (digitazione del testo).

Prima di esaminare altri, potenti comandi di WS, è bene che digitiate **ESATTAMENTE** quanto stiamo per dire, in modo da esser sicuri di seguire perfettamente le stesse indicazioni.

Cancellate tutto il testo presente in memoria (**Control + K + Q + S**) e, subito dopo la comparsa del menu principale, aprite un nuovo file (**D + ESEMPIO2**). Scrivete ora di seguito (correggendo con il tasto BackSpace eventuali errori di battitura) la seguente frase, interponendo **UN SOLO** spazio tra due parole successive (per ottenere i caratteri maiuscoli è sufficiente tener premuto uno dei due tasti Shift):

Commodore Computer Club è una rivista che è letta da molti lettori appassionati di computer.

Alla fine della digitazione premete il tasto Return, operazione che, in seguito, ci limiteremo ad indicare con **[R]**.

Se avete seguito **ALLA LETTERA** quanto suggerito, vi sarete accorti che, mentre stavate digitando la frase, giunti ad un certo punto, una parola è stata trasferita al rigo successivo!

Osservando lo schermo vi assale un primo attimo di smarrimento: che diavolo abbiamo combinato?

Ci pensa il computer

E' ora opportuno fare un salto indietro e richiamare ciò che abbiamo detto all'inizio: il computer considera, ciò che digitiamo, come un **insieme continuo** di caratteri alfanumerici, che alloca (= trascrive) nella sua memoria, carattere dopo carattere. Per semplificare la vita agli utenti, tuttavia, i programmatori di WS (versione 5.5) hanno fatto in modo che ogni rigo del testo apparisse sempre "completo" e identico a come verrebbe stampato su carta. Ecco, quindi, spiegato il motivo dello strano comportamento del testo, che sembra sfuggirci come un'anguilla a mano a mano che digitiamo o cancelliamo.

Il carattere di minore (**<**) che *eventualmente* compare all'**estrema destra** del video indica che dopo l'ultima parola (presente in quel rigo) è stato premuto il tasto Return.

Effettueremo ora una simulazione video, vale a dire che chiederemo al computer di visualizzare il testo così come verrebbe riprodotto su carta, se decidessimo di stamparlo.

Per realizzare la simulazione video è necessario il comando **Control + O + P** (a patto, però, che possediate la scheda grafica).

Si noti che:

* il documento ha un aspetto piuttosto diverso da quello che presentava durante la digitazione.

* all'estrema destra non vi sono i caratteri che indicano la pressione del tasto return (<).

Con questa esperienza abbiamo imparato un importante comando di WS (la simulazione video) e compreso (ce lo auguriamo!) il modo di "ragionare" del nostro W/p: durante la digitazione di un testo, i margini destro e sinistro del video, in realtà, sono fittizi e non corrispondono sempre a ciò che vedremo su carta. In sede di "composizione" del testo (e cioè in simulazione video o in fase di stampa vera e propria), il documento in memoria viene ricomposto secondo canoni a noi familiari. Da ciò consegue una serie di raccomandazioni:

* E' inutile (anzi, è **dannoso**) dividere in sillabe le parole che giungono, in fase di battitura, alla destra **fisica** del rigo video (se necessario, l'operazione viene svolta automaticamente dal computer).

* E' indispensabile inserire **SEMPRE** uno spazio tra una parola e la successiva, anche se (ad esempio, dopo un segno di punteggiatura come il punto o la virgola) sembra che non ve ne sia bisogno (pena la fusione di due parole apparentemente separate).

Altre considerazioni

Supponiamo di avere ancora a disposizione del testo. Posizionatevi, con i tasti cursore, su un carattere qualsiasi, purchè vi siano altri caratteri **dopo** di esso; assicuratevi, inoltre, che NON compaia la scritta **Insert** sul primo rigo in alto (se, invece, dovesse comparire, premete il tasto Insert, fino a far scomparire la parola stessa).

Se, a questo punto, premete la **barra spaziatrice**, potete notare che cancellate i caratteri sui quali vi muovete.

Ciò dimostra un modo di agire completamente diverso da una macchina da scrivere; con quest'ultima, infatti, non succede nulla, e men che meno cancellate qualcosa se, mediante barra spaziatrice, "passate sopra" caratteri già scritti. Ne consegue che:

* Lo spazio, se **NON** usato in modalità Insert, è, per il computer, un carattere come **TUTTI GLI ALTRI**: per spostarsi senza lasciar traccia è necessario servirsi **ESCLUSIVAMENTE** dei tasti cursore.

Per evitare cancellazioni involontarie, pertanto, vi consigliamo di operare sempre in modo Insert, dal momento che tale modalità permette, durante la digitazione, di spostare a destra tutti i caratteri presenti alla destra del cursore stesso. Per spostare il cursore, senza cancellare il tetso, dovrete usare i tasti cursore.

Allo scopo di esaminare gli altri comandi di WS vi consigliamo di digitare una (lunga) frase qualunque e di applicare, meglio se più di una volta, i comandi che seguono:

Inser (inserimento):
in questo modo (parola Inser presente in alto, sul primo rigo) se digitate qualcosa, ciò che è presente alla destra del cursore viene spostato.

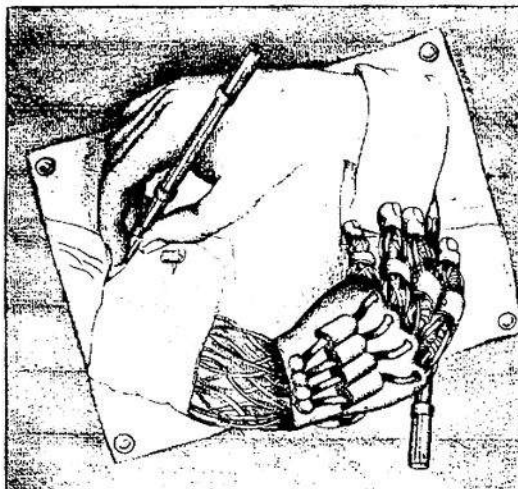
Delete (cancellazione):

* posizionate il cursore sul **primo** dei caratteri che volete cancellare.

* premete **Control + K + B**;

* spostate il cursore, mediante i tasti cursore, posizionandovi sull'**ultimo** carattere che volete che sia cancellato; premete **Control + K + K**: noterete che il gruppo di caratteri cambia colore.

* premete **Control + K + Y**: il gruppo di parole (o caratteri) evidenziato in reverse (scuro su fondo chiaro) verrà cancellato; a volte, nel caso di frasi troppo lunghe, compare una richiesta di conferma.

Se il numero di caratteri non è eccessivo, infatti, premendo **Control + U** potrete far riapparire quanto cancellato.

* il comando Delete va utilizzato in più occasioni: imparate ad usarlo con disinvoltura.

Control + S Save (registrazione): ogni volta che ricorrete a questo comando, il file attualmente attivo (di nome, ad esempio, LETTERA) verrà registrato, sovrapponendosi a quello vecchio. Quest'ultimo, comunque, sarà ancora presente sul disco, pur se con il suffisso modificato in **.BAK** (esempio: LETTERA.BAK); se quindi avete registrato un file combinando qualche pasticcio (esempio: dopo aver cancellato, per errore, una parte di testo troppo lunga da esser recuperata), potete interrompere il lavoro, "uscire" da WS, duplicare il nome del vecchio file (esempio: LETTERA.DUE), "rientrare" in WS, caricare LETTERA.DUE e proseguire nel lavoro.

E' indispensabile cambiar nome ai files dotati di suffisso **.BAK** perchè WS, per ovvi motivi di sicurezza, impedisce la modifica di tali documenti, pur se ne consente il "caricamento" allo scopo di esaminarli.

Control + K + T + nuovo nome: serve per registrare il documento presente in memoria con un altro nome. Con i comandi **Control + K + S** (oppure **Control + K + D**) il file in memoria viene sempre registrato con lo stesso nome con cui è stato caricato.

Control + K + R (Read = caricamento): è il comando per aggiungere un file-documento precedentemente registrato. Attenzione: il documento verrà caricato a partire dal punto in cui è presente il cursore.

L'eventuale parte di testo presente in memoria, posizionata dopo il cursore, verrà spostata in avanti per far spazio al file da caricare.

Control + Q + I (vai a pagina): il computer attende che indichiate la pagina su cui posizionare il cursore.

Control + Q + R Salta all'inizio del documento.

Control + Q + C Salta alla fine del documento.

Control + K + ' (accento) trasforma in minuscolo tutti i caratteri selezionati dalla precedente coppia di comandi (Control + K + B) e (Control + K + K) che, lo ricordiamo, appaiono in colore diverso.

Control + K + " (apici); idem come il precedente, solo che trasforma il blocco selezionato in maiuscolo.

Control + K + P (printer = stampa): invia su stampante il file successivamente indicato; questo apparirà nella forma riscontrabile con la simulazione video. Il menu che compare, prima della stampa, consente di indicare facilmente il numerto di copie desiderate, l'eventuale pausa a fine pagina e così via.

PgDn Visualizza la schermata successiva; **PgUp** visualizza la schermata precedente.

Control + Q + A: (ricerca e sostituisce): posizionare il cursore all'inizio della parte in cui si desidera effettuare la ricerca (tipicamente, all'inizio del testo); digitare, quando richiesto, la frase (o il gruppo di caratteri) che intendete ricercare. Alla successiva domanda (**Sostituire con:**) digitare il gruppo di caratteri che intendete sostituire. Sul terzo rigo potete rispondere con uno, o più, caratteri che rappresentano altrettanti comandi, tra cui: **G**, sostituzione globale; **N**, sostituire senza conferma; **U**, ignorare differenza tra caratteri minuscoli e maiuscoli; eccetera.

Esercizi

Consigliamo vivamente il lettore, per impratichirsi, di battere un testo qualunque e di applicare più volte i vari comandi in modo da raggiungere una certa esperienza e non commettere errori. Si consiglia di ricorrere spesso alla simulazione video (se disponete, ovviamente, della scheda grafica) per verificare se ciò che è stato battuto corrsponde a ciò che si immagina venga riprodotto.

In pratica, vi suggeriamo di:

* verificare SEMPRE la presenza della parola Inser.

* premere SEMPRE il tasto Return quando si decide di andare a capo; non premerlo MAI, invece, se giungete in prossimità della fine del rigo video: ci pensa il computer ad andare **automaticamente** a capo, che abbiate impostato la divisione automatica in sillabe oppure no.

* volendo cancellare un carattere, posizionare il cursore alla sua DESTRA e premere il tasto BackSpace tenendo presente che il carattere di spazio e Return sono considerati eguali a tutti gli altri.

* Ricordarsi che, durante le cancellazioni, tutti i caratteri presenti alla destra del cursore vengono "trascinati" verso sinistra.

* cancellando, volontariamente o meno, un carattere di ritorno carrello [R], la frase successiva viene "fusa" con la precedente.

* imparate subito ad usare i comandi per cancellare intere porzioni di testo evitando l'uso intensivo (ma inefficiente) del tasto backspace.

* imparate a spostare, o riprodurre, porzioni di testo mediante Control + K + V oppure Control + K + C, dopo aver selezionato la parte di testo che interessa.

* in caso di difficoltà (se, ad esempio, "entrate" per errore in un menu), premete il tasto **Esc** oppure **Control + U**.

* controllate spesso, soprattutto in caso di difficoltà, eventuali messaggi che dovessero comparire, di solito presenti nella prima riga in alto.

La formattazione del testo

Abbiamo visto che è possibile visualizzare, o mandare su stampante, il testo digitato.

C'è un modo semplicissimo per alterare la forma finale del documento. Vedremo ora, uno per uno, i principali comandi possibili.

WS è un w/p che consente di impostare i vari formati di stampa in punti qualsiasi del documento. Questo fatto comporta una maggior facilità d'uso ma, in caso di errori di impostazione, la scrittura del testo potrebbe interrompersi in corrispondenza dell'errore riscontrato.

Ciò significa che è sempre necessario, prima di effettuare una copia su carta, realizzare una simulazione video (Control + O + P), in modo da rendersi conto della correttezza dell'aspetto finale del documento.

I comandi "punto"

Se, come **primo** carattere di un rigo video, digitate un punto (.) noterete che immediatamente, all'estrema destra, compare un punto di domanda (**?**). Con

WS, infatti, particolari comandi, tra cui la formattazione del testo, vengono impartiti digitando un punto come primo carattere di un qualsiasi rigo video. Naturalmente, non solo il punto **non** verrà stampato (nè visualizzato in fase di simulazione video), ma non verrà stampato nessun carattere eventualmente presente dopo il punto stesso, sullo stesso rigo.

In altre parole, il carattere "punto" (purchè, lo ripetiamo, sia il **primo** carattere di una riga video) non verrà riprodotto, ma indicherà al computer che i caratteri posti in seguito vanno interpretati come **comandi** di stampa e non come caratteri da stampare.

Per capire come utilizzare la formattazione, cancellate il testo (Control + K + Q + S) e digitate un nuovo testo, lungo almeno una schermata e contenente almeno tre Return **[R]**.

Impostate il modo Insert, posizionatevi all'inizio del testo (Control + Home) e battete ciò che segue:

.LM 10
.RM 30

Prima di andare oltre, è necessario informare WS che il documento contiene comandi aggiuntivi. Per questo motivo è bene formattare l'intero testo presente in memoria, mediante i comandi **Control + Q + U**.

Sul video, a questo punto, si genera un movimento quasi convulso; alla fine noterete che i comandi sono stati eseguiti. **.LM10**, infatti, significa : *"Da questo momento il testo che segue deve trovarsi a distanza di 10 caratteri dal margine sinistro (LM = Left Margin, appunto margine sinistro)"*.

Analogamente, **.RM30** significa, nel codice WS: *"Da questo momento il margine destro (RM = right margin) deve essere posizionato a distanza di 30 caratteri, sempre conteggiati a partire da sinistra"*.

Effettuando, ora, la simulazione video (Control + O + P), noterete che il testo, da quel punto in poi, viene visualizzato non a partire dalla prima colonna (come prima) ma dalla decima; analogamente l'ultima parola del rigo visualizzato non va oltre la trentesima colonna (conteggiata a partire da sinistra).

Per esercizio, provate a modificare a volontà i valori impostati, come ad esempio:

.LM5
.RM60
.LM1
.RM20
...eccetera.

Noterete come, con una minima modifica, è possibile far cambiare aspetto al documento **senza la necessità di ribatterlo nuovamente**, come invece dovreste fare se usaste una comune macchina da scrivere! Inoltre, speriamo, avete capito perchè non bisogna **mai** premere il tasto Return alla fine del rigo fisico del video; in caso di ri-formattazione del testo, la potenzialità offerta dal computer, di allineare il documento, verrebbe limitata in modo considerevole.

In altre parole: premete il tasto Return **se, e solo se**, avete la necessità di "andare" a capo; cioè **solo** quando dovete iniziare un nuovo paragrafo.

Altri comandi di formattazione sono:

.LS 2 interlinea (2, nel caso specifico).

.PL 20 Lunghezza pagina (foglio di 20 righe, nel caso specifico). Consente di stabilire la lunghezza fisica del foglio di carta che verrà introdotto nella stampante. Tale indicazione è utile dal momento che il computer, automaticamente, inserirà un certo numero di righe vuote al momento in cui ritiene di esser giunto alla fine fisica del foglio.

.PA Cambio forzato pagina (il testo successivo al comando andrà stampato all'inizio di una pagina nuova).

.OJ1 Giustificazione del testo: i caratteri presenti alla destra del testo sono allineati verticalmente come quelli di sinistra; questa è l'impostatzione di default, appena si fa partire WS.

.OJ0 Disattiva la giustificazione. Da questo punto in poi il testo non viene allineato a destra, ed il risultato è quello che si otterrebbe con una comune macchina da scrivere. Tale modalità è detta, comunemente, "a bandiera".

.OC ON/OFF Centra il testo eliminando automaticamente l'eventuale giustificazione.

.H1 Messaggio (:**H**eader = intestazione). All'inizio di ogni pagina del documento, trascrive la parola *"Messaggio"* al centro del primo rigo in alto.

.F1 Messaggio (:**F**= Foot, piede) Come per .H1, solo che la parola *"messaggio"* viene inserita sulla prima riga in basso di ogni foglio.

Altri "comandi punto" sono descritti (oltre che sul manuale originale...) digitando un punto all'inizio di un rigo e, subito dopo, premendo il tasto **F1**. La schermata di aiuto che compare è solo la prima di numersose altre che descrivono tutti i comandi punto disponibili. Con un po' di pazienza, ed effettuando esperimenti, sarà possibile comprendere pienamente le potenzialità offerte da ciascuno di essi.

Una considerazione vitale

Molto spesso, lavorando con un word processor, quindi anche con WS, l'aspetto finale del documento non è quello che ci si aspetta, e non si riesce a capire il perchè.

Non ci riferiamo, ovviamente, agli errori di impostazione o di formattazione, ma ad altre imprecisioni che influiscono quasi esclusivamente sull'**apetto estetico** finale. Ci riferiamo, tra l'altro, ad un numero inspiegabilmente elevato di spazi tra due parole successive, a segni di punteggiatura mal distribuiti, a righe troppo piene seguite da altre troppo vuote.

Il motivo di tali incongruenze, nella quasi totalità dei casi, è dovuto alla mancanza di... comprensione, da parte nostra, nei riguardi del computer.

Per un calcolatore, infatti, una parola è rappresentata da un gruppo di caratteri, di qualsiasi natura, ravvicinati tra loro.

Analogamente, la separazione tra due parole è individuata, dal computer, **sempre e soltanto** dallo **spazio**: il calcolatore non sa (nè può saperlo) che un segno di punteggiatura separa due parole tra loro.

Ad esempio, il termine "`primo,secondo`" (in cui, cioè, non vi siano spazi nè prima nè dopo la virgola) è considerato un'**unica parola** benchè noi riusciamo a distinguere i due termini grazie alla presenza della virgola. Ciò comporta una conseguenza notevole in sede di formattazione del testo; se, infatti, la "parola" `primo,secondo` capita al termine di un rigo, e non può essere ospitata per intero a causa della sua lunghezza, viene sposatata PER INTERO all'inizio del rigo successivo, costringendo il computer ad inserire, sul rigo precedente, una gran quantità di spazi bianchi per pareggiare la giustificazione!

E' pertanto indispensabile far capire al calcolatore dove **inizia** e dove **termina** il

gruppo di caratteri che deve considerare come un'unica **parola**. Qui di seguito diamo, pertanto, gli accorgimenti da seguire SEMPRE per evitare errori di formattazione che influiscano sull'estetica del testo.

Primo gruppo

Caratteri che devono essere **attaccati** alla parola che li precede e **separati** da quella che li segue:

Punto (**.**) punto e virgola (**;**) virgola (**,**) doppio punto (**:**) punto esclamativo (**!**) punto interrogativo (**?**) percentuale (%) accento o apostrofo (') chiusura di apici (**"**) parentesi tonda chiusa **)** parentesi quadra chiusa **]** eguale (=)

Esempi: bisogna scrivere...

`Parola, parola; parola!`

(notare lo spazio **dopo** il segno di punteggiatura)...e non...

`Parola , parola ; parola !`

...e non...

`Parola ,parola ;parola !`

...oppure...

`Parola,parola;parola!`

Secondo gruppo

Caratteri che devono essere **attaccati** alla parola che li segue e **separati** da quella che li precede:

Apertura di apici (") parentesi tonda aperta **(** parentesi quadra aperta (**[**)

Esempi:

`Aereo, cioè "oggetto che vola".`

...e non...

`Aereo, cioè " oggetto che vola "`

...e non...

`Aereo,cioè"oggetto che vola"`

Conclusioni

Molti altri potenti comandi sono disponibili con WS, tanto che il manuale di istruzioni, disponibile anche in italiano, è piuttosto voluminoso e ricco di esempi. Coloro che volessero saperne di più, o segnalare difficoltà di utilizzo, potranno telefonarci, o scriverci segnalando casi particolari non individuabili sul manuale.

Acquistare il package originale, reperibile presso i rivenditori autorizzati, sarà comunque la **soluzione migliore** per sfruttare al massimo quello che può esser definito come uno dei migliori strumenti software per la gestione di testi.

L'esempio riportato

Nella pagina che segue (rimpicciolita in modo vergognoso al solo scopo di risparmiare spazio), il primo canto della **Divina Commedia**, sufficientemente noto a tutti noi(!), è stato digitato di seguito, **senza**, cioè, premere il tasto Return alla fine di ogni verso, ma solo alla fine di ogni "paragrafo".

Nelle varie parti in cui il canto è suddiviso, sono stati introdotti alcuni "comandi punto" allo scopo di verificare l'aspetto finale del documento. Lo studio approfondito dell'esempio sarà prezioso per comprendere il modo di "ragionare" di Word Star al momento in cui incontra i comandi speciali di formattazione del testo.

Parte 1

E' stato regolato il solo margine destro (**.RM 70**) ed impostata l'eliminazione della giustificazione (**.OJ Off**). Si noti, a destra, la parte frastagliata del testo che, di norma, si ottiene con una qualsiasi macchina da scrivere.

Parte 2

L'interlinea è stata fissata al valore 2 (**.LS 2**)

Parte 3

L'interlinea viene riportata al valore standard (**.LS 1**).

Parte 4

La giustificazione (**.OJ On** oppure **.OJ1**) impedisce il frastagliamento: a sinistra e a destra il testo risulta allineato **verticalmente** grazie all'inserimento automatico di un certo numero di spazi vuoti.

Parte 5

Vengono modificati i margini sinistro e destro (**.LM 10 .RM 40**). Le impostazioni precedenti rimangono confermate.

Parte 6

Viene fatto rientrare il testo di 3 caratteri (**.PM 13**). Il "rientro" corrisponde al numero di colonne conteggiate a partire dalla prima colonna di sinistra; in questa fase, infatti, il margine sinistro vale già 10; ne consegue che la rientranza sarà di 3 caratteri (13 - 10).

In seguito il comando verrà annullato (**.PM Off** oppure: **.PM 0**); in caso contrario il comando (come tutti i "comandi punto" impostati) rimarrebbe attivo fino alla fine del documento!

Parte 7

Viene imposta la centratura (**.OC On**) del testo che segue. I comandi precedenti, in contrasto con tale ordine, vengono quindi ignorati; i margini precedentemente impostati non "contraddicono" l'impostazione e rimangono quindi attivi.

Parte 8

Vengono modificati margini destro e sinistro. Il comando di centratura, però, rimane attivo; si noti, in particolare, l'ultima parola (**pria**) posta al centro del rigo.

Parte 9

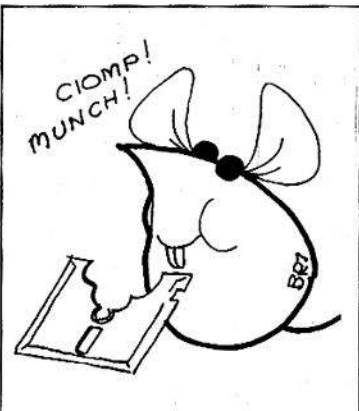
Una nuova impostazione del margine sinistro (**.LM 20**) non ha alcuna conseguenza per il margine destro e gli altri comandi precedentemente impartiti.

Parte 10

Viene annullato (**.OC OFF**) il comando di centratura. Ritorna quindi attivo il comando **.OJ On** (vedi parte 4).

Parte 11

Un'ultima modifica ai margini non implica alcuna novità.

.OJO
.RM70

LA DIVINA COMMEDIA

trascrizione di Maurizio Dell'Abate

Canto Primo

Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura, che la diritta via era smarrita. Ah quanto a dir qual era e' cosa dura esta selva selvaggia e aspra e forte che nel pensier rinnova la paura! Tant'e' amara che poco e' piu' morte; ma per trattar del ben ch'io vi trovai, diro' de l'altre cose ch'io v'ho scorte. Io non so ben ridir com'io v'entrai: tant'era pieno di sonno a quel punto che la verace via abbandonai.

.LS 2

Ma poi ch'i' fui al pie' d'un colle giunto, la' dove terminava quella

valle che m'avea di paura il cor compunto, guardai in alto, e vidi le

sue spalle vestite gia' de' raggi del pianeta che mena dritto altrui

per ogni calle. Allor fu la paura un poco queta, che nel lago del cor

m'era durata la notte ch'io passai con tanta pieta.

.LS1

E come quei che con lena affannata uscito fuor del pelago a la riva, si volge a l'acqua perigliosa e guata; cosi' l'animo mio, ch'ancor fuggiva, si volse a retro a rimirar lo passo che non lascio' gia' mai persona viva.

.OJ 1

Poi ch'ei posato un poco il corpo lasso, ripresi via per la piaggia diserta, si' che 'l pie' fermo sempre era 'l piu' basso. Ed ecco, quasi al cominciar de l'erta, una lonza leggiera e presta molto, che di pel maculato era coverta; e non mi si partia dinanti al volto, anzi impediva tanto il mio cammino, ch'i' fui per ritornar piu' volte volto. Temp'era dal principio del mattino, e 'l sol montava 'n su con quelle stelle ch'eran con lui quando l'amor divino mosse di prima quelle cose belle; si' ch'a bene sperar m'era cagione di quella fera a la gaetta pelle l'ora del tempo e la dolce stagione; ma non si' che paura non mi desse la vista che m'apparve d'un leone:

.LM 10
.RM 40

questi parea che contra me venesse con la test'alta e con rabbiosa fame, si' che parea che l'aere ne temesse; e d'una lupa, che di tutte brame sembiava carca ne la sua magrezza, e molte genti fe' gia' viver grame: questa mi porse tanto di gravezza con la paura ch'uscia di sua vista, ch'io perdei la speranza de l'altezza.

.PM 13

E qual e' quei che volentieri acquista, e giugne 'l tempo che perder lo face, che 'n tutt' i suoi pensier piange e s'attrista; tal mi fece la bestia sanza pace, che, venendomi incontro, a poco a poco mi ripigneva la' dove 'l sol tace. Mentre ch'i' ruinava in basso loco, dinanzi a li occhi mi si fu offerto chi per lungo silenzio parea fioco. Quando vidi costui nel gran diserto, "Miserere di me", gridai a lui, "qual che tu sii, od ombra od omo certo!

"Rispuosemi: "Non omo, omo gia' fui; e li parenti miei furon lombardi, mantovani per patria ambedui. Nacqui sub Julio, ancor che fosse tardi, e vissi a Roma sotto 'l buono Augusto al tempo de li dei falsi e bugiardi. Poeta fui, e cantai di quel giusto figliuol d'Anchise che venne da Troia, poi che il superbo Ilion fu combusto.

.PM 0
.OC ON

Ma tu perche' ritorni a tanta
noia? Perche' non sali il
dilettoso monte ch'e' principio
e cagion di tutta gioia?" "Or
se' tu quel Virgilio e quella
fonte che spandi di parlar si'
largo fiume?" rispuos'io lui
con vergognosa fronte. "O degli
altri poeti onore e lume,
vagliami il lungo studio e 'l
grande amore che m'ha fatto
cercar lo tuo volume.

.LM 1
.RM 70

Tu se' lo mio maestro e 'l mio autore; tu se' solo colui da cu' io
tolsi lo bello stilo che m'ha fatto onore. Vedi la bestia per cu' io
mi volsi: aiutami da lei, famoso saggio, ch'ella mi fa tremar le vene
e i polsi". "A te convien tenere altro viaggio", rispuose poi che
lagrimar mi vide, "se vuo' campar d'esto loco selvaggio; che' questa
bestia, per la qual tu gride, non lascia altrui passar per la sua via,
ma tanto lo 'mpedisce che l'uccide; e ha natura si' malvagia e ria,
che mai non empie la bramosa voglia, e dopo il pasto ha piu' fame che
pria.

.LM 20

Molti son li animali a cui s'ammoglia, e piu'
saranno ancora, infin che 'l Veltro verra', che la
fara' morir con doglia. Questi non cibera' terra
ne' peltro, ma sapienza, amore e virtute, e sua
nazion sara' tra feltro e feltro. Di quella umile
Italia fia salute, per cui mori' la vergine
Cammilla, Eurialo e Turno e Niso di ferute. Questi
la caccera' per ogni villa, fin che l'avra'
rimessa ne lo 'nferno, la' onde invidia prima
dipartilla.

.OC OFF

Ond'io per lo tuo me' penso e discerno che tu mi segui, e io saro' la tua guida, e trarrotti di qui per luogo etterno; ov'udirai le disperate strida, vedrai li antichi spiriti dolenti, che la seconda morte ciascun grida; e vedrai color che son contenti nel foco, perche' speran di venire quando che sia a le beate genti. A le qua poi se tu vorrai salire, anima fia a cio' piu' di me degna:

.LM 1
.RM 50

con lei ti lascero' nel mio partire; che' quello impera' nel mio partire; che' quello imperador che la' su regna, perch'io fu' ribellante a la sua legge, non vuol che 'n sua citta' per me si vegna. In tutte parti impera e quivi regge; quivi e' la sua citta' e l'alto seggio: oh felice colui cu' ivi elegge! " E io a lui: "Poeta, io ti richeggio per quello Dio che tu non conoscesti, accio' ch'io fugga questo male e peggio, che tu mi meni la' dov'or dicesti, si' ch'io veggia la porta di San Pietro e color cui tu fai cotanto mesti". Allor si mosse, e io lo tenni retro.

di Mario Saggese

GIOCARE AL LOTTO, MA CON PRUDENZA...

La passione per la programmazione, più che per il gioco in se stesso, ha spinto un nostro affezionato e validissimo lettore a scrivere uno straordinario programma in "C" che la Systems offre solo per corrispondenza

Il lunghissimo programma (più di **42 Kbytes** di codice sorgente in linguaggio **"C"**!) può essere di aiuto agli appassionati del gioco del Lotto, perché permette di ottenere le numerose informazioni che più comunemente vengono ricercate per compilare le giocate.

Non suggerisce pronostici, ma fornisce indicazioni circa i molti tipi di ritardi che possono verificarsi nell'uscita dei numeri o di loro particolari abbinamenti. Ognuno può costruirsi le proprie giocate, secondo i propri gusti o abitudini, non trascurando la necessaria moderazione e prudenza.

Il programma è stato realizzato, per Computer **Ms - Dos** compatibili, in linguaggio **Turbo C Borland**, che consente una notevole velocità di elaborazione.

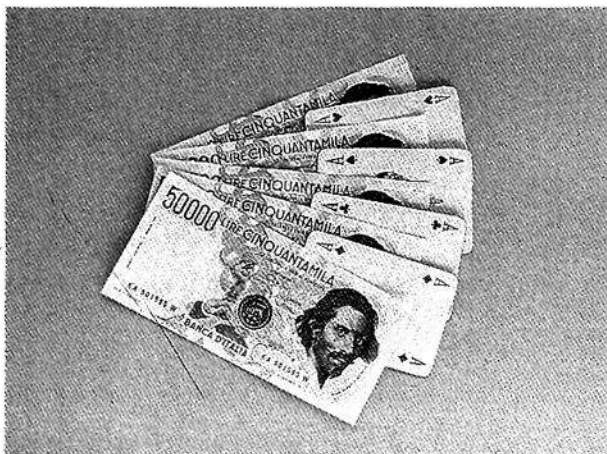
Ne riassumiamo, ora, le modalità d'uso, peraltro molto semplici, a patto che siano rispettate le condizioni segnalate alla conclusione dell'esame dell'opzione numero 8 (immissione settimanale dei numeri estratti, che costituisce il fondamento di tutte le elaborazioni).

Impiego iniziale

Quando vi procurerete il dischetto (in formato **5.25** oppure **3.5** pollici, a vostra scelta) dovrete caricare il sistema operativo dal drive A; sistemare il dischetto contenente il programma nel drive B e rendere questo drive attivo, digitando **B:** seguito dalla pressione del tasto Return.

Digitate **LOT** seguito da Return. Se si possiede un disco rigido, occorre copiare su questo tutti i files e al prompt **(C>)**

```
    Pgrogramma elaborato
    con Turbo C da M.Saggese
    Estrazioni settimanali in
    archivio 158: ultima 05/01/91

1..Ritardi maggiori/Frequenze minori
2..Ritardi/Frequenze d'ogni Ruota
3..Ritardi numeri singoli
4..Rit. decine/cadenze/gemelli/vertibili
5..Numeri con ritardi particolari
6..Ricerca di ambi usciti
7..Ricerca  estrazioni
8..Aggiornamento Estrazioni/Ritardi
9..Correzione Ritardi/Estrazioni
Indica scelta : <esc> chiude
```

La schermata di presentazione

Ritardi maggiori di tutte le Ruote al 05/01/91

Bari	*	5	34	36	45	12	1	29	61	71	68
Ritardi		57	48	46	44	43	41	40	40	39	38
Cagliari	*	44	42	10	32	24	46	11	89	41	16
Ritardi		114	73	68	67	64	57	54	53	51	46
Firenze	*	55	75	68	10	90	51	37	20	1	79
Ritardi		99	83	80	70	69	56	44	43	41	40
Genova	*	64	17	90	66	23	8	67	57	70	19
Ritardi		80	64	63	53	46	43	36	32	30	29
Milano	*	5	36	33	22	87	34	15	1	56	28
Ritardi		114	104	76	71	68	61	60	50	45	43
Napoli	*	23	66	57	55	15	62	72	82	89	84
Ritardi		78	75	63	61	60	60	55	55	49	44
Palermo	*	61	73	38	87	33	10	23	31	77	80
Ritardi		94	65	61	55	39	37	37	37	36	35
Roma	*	46	59	47	16	49	7	20	30	3	58
Ritardi		91	81	63	55	54	49	49	46	45	40
Torino	*	83	26	23	29	24	11	16	25	4	63
Ritardi		79	70	68	57	56	52	45	43	42	39
Venezia	*	69	90	26	34	80	23	15	28	17	8
Ritardi		107	66	62	49	45	43	42	42	35	34

Frequenze minori dal 02/01/88 al 05/01/91

Bari	*	2	36	3	5	16	20	34	51	61	74
Frequenze		3	4	5	5	5	5	5	5	5	5
Cagliari	*	30	42	45	89	2	27	44	46	57	59
Frequenze		2	2	4	4	5	5	5	5	5	5
Firenze	*	54	35	49	52	55	90	69	71	75	81
Frequenze		2	4	4	4	4	4	5	5	5	5
Genova	*	15	17	60	72	79	9	34	48	52	82
Frequenze		4	4	4	4	4	5	5	5	5	5
Milano	*	47	15	17	5	7	9	10	33	34	63
Frequenze		3	4	4	5	5	5	5	5	5	5
Napoli	*	55	65	82	43	60	58	62	9	23	38
Frequenze		3	3	3	4	4	5	5	6	6	6
Palermo	*	77	48	66	73	3	10	24	52	61	62
Frequenze		3	4	4	4	5	5	5	5	5	5
Roma	*	7	16	49	69	85	30	14	26	37	46
Frequenze		3	3	3	3	3	4	5	5	5	5
Torino	*	2	26	27	8	11	15	71	77	84	1
Frequenze		4	4	4	5	5	5	5	5	5	6
Venezia	*	13	90	69	26	31	21	34	62	73	79
Frequenze		2	2	3	4	4	5	5	5	5	5

digitare LOT. Come consuetudine della Systems Editoriale, teniamo a sottolineare che il dischetto è privo di qualsiasi protezione; anzi, precisiamo che sul dischetto è addirittura presente il listato **sorgente**; chiunque, pertanto, potrà introdurre le modifiche che desidera, dopo averlo esaminato con calma.

Il programma parte immediatamente e compare sullo schermo la richiesta della scelta tra visione a colori (ovviamente se consentita dal computer posseduto) o in bianco e nero. Questa richiesta, nell'uso successivo del programma, non verrà più formulata in quanto la scelta iniziale viene memorizzata in un breve file su disco.

Compaiono quindi il numero delle estrazioni già registrate, con la data dell'ultimo aggiornamento effettuato, oltre alla indicazione del linguaggio usato e al nome dell'autore (Mario Saggese). Vengono quindi caricati, da disco, nella memoria del computer alcuni dati d'uso ed è subito presentato il menu principale, comprendente **nove opzioni numerate**. La scelta si effettua digitando il numero di quella desiderata. Completato lo sviluppo dell'opzione, ricompare il menu, per eventuali altre scelte.

Alcune di queste, per essere costituite da più di due voci, comportano anche menu secondari, che vedremo meglio in seguito. La chiusura del programma si ottiene premendo il tasto **Esc**. Per chiarezza espositiva tratteremo le opzioni ed i risultati conseguibili con ognuna di esse, peraltro chiaramente indicate nelle rispettive denominazioni.

Unica accortezza da porre, comune a tutte, riguarda la risposta da dare alle domande formulate nel corso del programma. Questa deve sempre corrispondere ad una delle lettere, volta per volta indicate, e non ad altre. I caratteri possono essere introdotti sia in minuscolo che in maiuscolo.

Ritardi / Frequenze

La prima opzione ordina i ritardi di uscita (in numero di settimane trascorse dall'ultima estrazione) dei numeri di ogni ruota e presenta, in un prospetto generale, i dieci numeri che presentano maggiori ritardi, con le rispettive settimane di assenza.

E' questa la più comune valutazione dei ritardi, segnalata anche da **Televideo** (limitatamente ai primi 5 maggiori ritardatari). Naturalmente, ad ogni estrazione settimanale, si modifica l'entità dei ritardi per i vari numeri, con l'azzeramento per quelli usciti e l'incremento di una unità per tutti gli altri. Per questo motivo, nel prospetto, figura **la data** alla quale i dati si riferiscono ed entro la quale sono, pertanto, da considerare validi. Con la immissione di nuove estrazioni (opzione 8), essi sono regolarmente aggiornati e contemporaneamente i numeri usciti sono registrati nell'apposito archivio (dal nome: **Estraz.lot**).

Come è visibile nel menu, l'opzione n. 1 permette anche la ricerca dei numeri meno spesso usciti, nell'intervallo di tempo compreso tra la prima e l'ultima settimana registrate (i ritardi hanno, invece, valori assoluti).

Anche le frequenze si modificano ogni settimana (aumentano, ovviamente, di una unità quelle dei soli numeri usciti).

Ruota Roma		Numero	Ritardo al 05/01/91								
1	13	16	55	31	0	46	91	61	12	76	2
2	2	17	7	32	6	47	63	62	13	77	37
3	45	18	4	33	34	48	21	63	1	78	19
4	2	19	0	34	19	49	54	64	14	79	0
5	6	20	49	35	8	50	1	65	2	80	23
6	33	21	0	36	8	51	9	66	4	81	10
7	49	22	14	37	24	52	36	67	3	82	0
8	12	23	9	38	1	53	7	68	28	83	4
9	14	24	15	39	3	54	17	69	3	84	23
10	16	25	1	40	5	55	5	70	3	85	10
11	20	26	4	41	6	56	2	71	6	86	27
12	36	27	9	42	10	57	6	72	4	87	8
13	26	28	32	43	16	58	40	73	12	88	22
14	25	29	11	44	29	59	81	74	8	89	22
15	7	30	46	45	10	60	15	75	3	90	5

Ruota Roma		Numero	Frequenza dal 02/01/88 al 05/01/91								
1	8	16	3	31	8	46	5	61	12	76	12
2	9	17	13	32	6	47	5	62	8	77	12
3	11	18	11	33	7	48	5	63	8	78	8
4	7	19	8	34	8	49	3	64	10	79	11
5	9	20	8	35	7	50	13	65	14	80	8
6	7	21	15	36	8	51	10	66	8	81	12
7	3	22	10	37	5	52	11	67	12	82	8
8	13	23	7	38	7	53	9	68	8	83	8
9	6	24	8	39	9	54	6	69	3	84	7
10	16	25	9	40	11	55	8	70	13	85	3
11	8	26	5	41	11	56	10	71	11	86	13
12	10	27	8	42	12	57	9	72	16	87	10
13	7	28	9	43	9	58	5	73	10	88	14
14	5	29	13	44	11	59	7	74	8	89	11
15	7	30	4	45	8	60	6	75	5	90	11

Somma frequen.:		Numero	Frequenza dal 02/01/88 al 05/01/91								
1	77	16	64	31	100	46	75	61	91	76	97
2	86	17	80	32	86	47	89	62	91	77	81
3	82	18	86	33	86	48	81	63	86	78	95
4	92	19	94	34	77	49	83	64	92	79	92
5	85	20	78	35	76	50	82	65	88	80	96
6	102	21	93	36	85	51	81	66	90	81	89
7	90	22	95	37	82	52	92	67	89	82	84
8	89	23	91	38	85	53	109	68	86	83	94
9	73	24	72	39	94	54	90	69	83	84	91
10	96	25	79	40	105	55	78	70	96	85	93
11	89	26	73	41	78	56	94	71	90	86	83
12	87	27	104	42	88	57	88	72	98	87	91
13	87	28	96	43	89	58	82	73	100	88	104
14	81	29	98	44	81	59	95	74	100	89	98
15	76	30	78	45	87	60	80	75	85	90	76

Invocando questa opzione verrà, pertanto, posta la domanda se si desiderano i ritardi maggiori o le frequenze minori.

Con **R** compariranno sullo schermo i ritardi; con **F**, invece, le frequenze. La pressione di un qualsiasi altro tasto non ha effetto. Concludiamo l'esame della prima opzione, avvertendo che i dati del prospetto possono essere inviati su stampante, digitando la lettera **P**.

Ruota per Ruota

Con la scelta n. 2 del menu principale è possibile osservare, raccolti in un unico prospetto per ogni Ruota, i ritardi (**R**) o le frequenze (**F**), relativi ai 90 numeri. La selezione è attuabile in maniera identica a quella della prima opzione, con l'avvertenza che i dati per i Ritardi sono assoluti, mentre quelli per le Frequenze sono relativi, come già detto, alle sole estrazioni registrate.

La scelta della Ruota, che si desidera esaminare, si effettua digitando le prime due lettere del nome (**ro** per Roma, **na** per Napoli, eccetera.).

Solo per le frequenze, infine, per scopi statistici, è possibile ottenere anche un prospetto, con la somma delle frequenze di tutti i numeri per le dieci Ruote.

Ritardi

L'opzione n. 3 consente di evidenziare, per numeri singoli, i ritardi accumulati in ogni Ruota, con somma finale di tutti i ritardi.

```
Ritardi al 05/01/91
Numero 45          :
:
Bari        44
Cagliari    16
Firenze      3
Genova       0
Milano      17
Napoli       6
Palermo      2
Roma        10
Torino      29
Venezia      6
*Totale    133
Tasti a=avanti
z=indietro
<CR> esce
```

Ritardi in base alle decine al 05/01/91

:	0	1	2	3	4	5	6	7	8
Bari	4	5	14	12	13	3	0	1	2
Cagliari	0	1	9	8	16	10	4	24	10
Firenze	2	18	5	15	8	6	13	0	5
Genova	11	6	8	1	0	4	32	2	1
Milano	38	19	4	11	13	2	1	0	9
Napoli	0	1	5	5	15	23	0	6	24
Palermo	21	16	3	5	1	2	10	9	4
Roma	2	7	9	8	10	9	3	3	10
Torino	13	13	35	3	2	5	10	1	8
Venezia	8	7	6	18	9	1	11	4	31

Ritardi in base alle cadenze al 05/01/91

:	0	1	2	3	4	5	6	7	8	9
Bari	16	14	0	7	2	9	3	19	12	14
Cagliari	4	20	10	15	22	3	17	5	31	13
Firenze	16	5	0	1	7	13	0	2	91	25
Genova	20	6	5	3	1	0	19	11	10	2
Milano	4	9	3	4	1	6	14	0	9	6
Napoli	22	3	25	8	37	44	24	1	4	54
Palermo	37	0	14	29	3	5	1	8	11	12
Roma	5	0	15	12	17	7	2	9	40	0
Torino	17	19	11	4	25	7	45	2	5	4
Venezia	3	8	4	25	27	5	17	2	40	2

Per passare da un numero al successivo (o al precedente) senza tornare al menu, va premuto il tasto **A**, (oppure il tasto **Z**). La pressione di ogni altro tasto vi fa tornare al menu principale. L'opzione, presente in quasi tutti i programmi sul Lotto, permette di scorrere nei due sensi tutti i numeri, agevolando confronti o ripensamenti ed evitando il ricorso continuo al menu per richieste specifiche.

Abbinamenti

La quarta opzione riguarda, a differenza delle precedenti, ritardi di numeri abbinati per particolari caratteristiche, ben note agli appassionati del Lotto. Sono state scelte, tra le altre, le quattro più diffuse: l'appartenenza di due numeri alla stessa **decina**; identica cifra terminale (**cadenza**), eguaglianza di cifre (**gemelli**); cifre invertite (**vertibili**).

Per ognuna delle **9** decine o delle **10** cadenze possibili, e per ogni Ruota, i ritardi sono calcolati in base all'uscita di numeri appartenenti alla stessa decina o aventi la stessa cadenza, indipendentemente dal valore dei numeri.

Si afferma, ad esempio, che la decina 10 ritarda da 20 settimane su una determinata Ruota, se mancano da 20 settimane due qualsiasi dei numeri appartenenti alla decina 10 (da 10 a 19).

Per i gemelli vengono inizialmente presentati due prospetti affiancati: nel primo figurano, per ogni Ruota, le settimane di assenza di estrazioni di qualsiasi coppia di gemelli. Nel secondo sono indicate, per ogni coppia, i ritardi maggiori per tutte le Ruote, prendendo come parametro di valutazione l'ultima sortita della coppia su una qualsiasi Ruota.

Ai due prospetti generali possono seguire, a richiesta, altri due prospetti analitici nei quali, per ciascuna Ruota, sono mostrati i ritardi delle ventotto coppie di gemelli possibili.

Per i **vertibili** la procedura (ed i prospetti) sono analoghi a quelli detti per i gemelli (naturalmente i due numeri vertibili prendono il posto di una coppia di gemelli).

L'opzione si avvale, per la scelta, di un menu secondario. In tale menu la pressione della lettera iniziale (**d** per decine, **c** per cadenze, **g** per gemelli, **v** per vertibili), manda in esecuzione la routine relativa, secondo una consuetudine comune ad altri programmi. Alla conclusione di ogni routine, il menu secondario viene ripresentato, per eventuali nuove scelte.

Precisiamo che l'indicazione delle decine, nel prospetto, è limitata alla cifra della decina (**0** per numeri da 1 a 9, **1** per numeri da 10 a 19, **2** per numeri da 20 a 29 e così via). Per le cadenze viene invece indicata la sola cifra finale (**0** per numeri che terminano con 0, **1** per quelli terminanti con 1 e così via). Per ogni

Settimane di assenza di Vertibili al 05/01/91				Vertibili con piu' ritardi su tutte le Ruote		
Ruote		Settimane	: :	Vertibili		Ritardi
Bari	->	1	: :	4/8	->	145
Cagliari	->	2	: :	7/8	->	140
Firenze	->	7	: :	2/6	->	77
Genova	->	56	: :	1/4	->	71
Milano	->	7	: :	5/7	->	65
Napoli	->	42	: :	3/6	->	62
Palermo	->	2	: :	1/6	->	57
Roma	->	2	: :	3/7	->	56
Torino	->	1	: :	2/8	->	52
Venezia	->	25	: :	3/8	->	49

coppia di gemelli viene indicata, per ogni membro, una delle cifre uguali (1/2 equivale a 11/22, 3/4 a 33/44, 6/7 a 66/77, eccetera).

Per i **vertibili**, che sono presentati in modo analogo ai gemelli, basta invertire le cifre (1/2 = 12 . 21; 3/4 = 34 . 43; 6/7 = 67 . 76, eccetera). La particolare forma di visualizzazione è dovuta alla capienza dello schermo; consente peraltro di far risparmiare codice macchina in fase di compilazione e diventa di interpretazione agevole dopo un breve uso del programma.

Anche per i ritardi di decine, cadenze, gemelli e vertibili è prevista copia su carta.

Ritardi particolari

Definita come "Numeri con ritardi particolari" l'opzione visualizza, in due schermate successive, i numeri che presentano le più alte somme di ritardi verificatisi su tutte le Ruote e quelli assenti da 1 a più settimane, sempre su tutte le Ruote.

Le due routines sono autoesplicative e non necessitano di particolari commenti, se si esclude l'interesse che taluni pongono soprattutto alla seconda.

Opzione numero 6

Questa opzione, per esser considerata significativa, ha bisogno di una ampia base di dati. Con quelli a disposizione (estrazioni finora registrate) si ottengono risultati già buoni, che evidentemente miglioreranno ancora con l'aggiornamento settimanale dei dati, da compiere a cura dell'utente stesso.

Essa permette, attraverso l'esame dell'intero archivio delle estrazioni, la ricerca degli ambi estratti secondo due modalità e, inoltre, la individuazione dei ritardi degli ambi stessi, ancora con due modalità.

Con la scelta **A**, indicati due dei novanta numeri, viene ricercata l'eventuale uscita contemporanea, per ogni Ruota. In caso positivo è segnalata, per le ruote interessate, oltre alla data di estrazione (l'ultima, nel caso siano più di una), anche il numero d'ordine con cui la settimana in questione è presente nell'archivio.

La seconda modalità (scelta **N**) riguarda ciascuna Ruota, isolatamente: indicato un numero, vengono ricercati, tra gli altri 89, quelli con cui ha formato ambo sulla ruota scelta, sempre nell'arco di tempo di tutte le settimane registrate.

Nei casi positivi è indicata la settimana di estrazione (anche qui l'ultima, ove siano più di una).

Per quel che riguarda i ritardi su tutte le ruote, con la prima modalità (scelta **R**) vengono presentati gli ambi con i ritardi maggiori; con la seconda (scelta **S**) le settimane di assenza di ambi specifici, richiedibili dall'utente; ribadiamo, ancora

Settimane di assenza dei numeri seguenti su tutte le Ruote al 05/01/91:

Sett.	Numeri
Sett. 1:	10 13 14 16 23 25 29 34 38 40 41 47 53 56 58 59 64 65 66 67 71 78 84 86
Sett. 2:	5 7 44 57 73 81 83
Sett. 3:	8 15 24 32 43 60 70 89
Sett. 4:	12 18 20 26 55 61 80 88
Sett. 5:	11 74
Sett. 6:	1 22
Sett. 7:	nessun numero
Sett. 8:	28

una volta, che i limiti dell'indagine si situano tra la prima e l'ultima settimana archiviate.

Anche per i ritardi è indicato, oltre al valore del ritardo, il numero d'ordine di registrazione della settimana, in cui gli ambi sono usciti per l'ultima volta. La scelte relative alle ricerche o ai ritardi, si effettuano tramite un menu secondario, funzionalmente analogo a quello già accennato per l'opzione 4 del menu principale.

Opzione numero 7

La routine collegata a questa opzione permette di esaminare tutti i numeri di una determinata estrazione. Le estrazioni sono, come già detto, registrate in

Settimane di assenza di Gemelli al 05/01/91				Gemelli con piu' ritardi su tutte le Ruote			
Ruote		Settimane	: :	Gemelli		Ritardi	
Bari	->	17	: :	3/8	->	115	:
Cagliari	->	33	: :	4/5	->	113	:
Firenze	->	58	: :	1/6	->	106	:
Genova	->	22	: :	6/8	->	78	:
Milano	->	24	: :	2/3	->	66	:
Napoli	->	26	: :	2/5	->	61	:
Palermo	->	4	: :	3/7	->	59	:
Roma	->	18	: :	2/6	->	50	:
Torino	->	17	: :	1/2	->	49	:
Venezia	->	12	: :	1/5	->	40	:

```
Ambo 43/78
             sett. ultima uscita
Bari             ----------
Cagliari         ----------
Firenze    sett.121  21/04/90
Genova           ----------
Milano           ----------
Napoli           ----------
Palermo          ----------
Roma             ----------
Torino           ----------
Venezia sett. 38  17/09/88
Ricerca dal 02/01/88
             al 05/01/91
```

un archivio su disco e da questo recuperabili in ogni momento.

Ogni estrazione occupa nell'archivio **Estraz.lot** una determinata posizione che può essere individuata tramite un numero d'ordine. Indicando il numero della settimana scelta, questa sarà estratta dal file e mostrata.

A richiesta viene presentato un elenco completo delle estrazioni registrate, con i numeri d'ordine e le date relative. Lo scorrimento dell'elenco su video può essere interrotto quando si vuole.

Come per i ritardi dei numeri singoli è possibile percorrere l'intero archivio in senso ascendente o discendente.

E' necessario indicare solo inizialmente la settimana da cui partire.

Opzione numero 8

A questa opzione corrisponde la parte più importante e impegnativa del programma. Ad essa è affidato, infatti, il compito di memorizzare ogni settimana i numeri dell'estrazione in atto, registrarli su disco e procedere alle inevitabili modifiche statistiche sui ritardi e sugli altri dati coinvolti. L'immissione dei numeri è guidata dal computer, che indica in successione le dieci Ruote, con la richiesta di immettere i cinque numeri di ogni ruota. Un segnale acustico avverte su eventuali errori di digitazione (numeri fuori range minori di 1 o superiori a 90 oppure ripetuti).

Al termine dell'immissione viene presentato un quadro completo dei dati per eventuali correzioni, anche multiple. Queste sono effettuate, per un miglior controllo, sotto gli occhi stessi dell'utilizzatore (il numero esatto prende, nel prospetto, il posto di quello errato). Prima di procedere, infine, alle registrazioni e alle modifiche dei ritardi, viene chiesta conferma. Si tratta, come è in conclusione evidente, della parte più ponderosa e più ricca di conseguenze dell'intero programma. Di qui la predisposizione di forme accurate di controllo, nelle fasi di sviluppo della routine.

Due raccomandazioni importanti vanno fatte a tal proposito:

A) Per un corretto funzionamento del programma nessuna estrazione deve mancare nell'archivio tra l'ultima registrata (la data compare sempre nella schermata introduttiva) e quella in corso. Se ne mancano alcune, queste possono essere inserite e registrate, anche una dopo l'altra, rispettando però rigidamente l'ordine con cui esse si sono succedute nel tempo. Particolare attenzione va posta, sotto questo profilo, soprattutto all'atto del primo impiego del programma.

B) Il programma prevede, come si vedrà nei dettagli nell'esame della prossima opzione, la possibilità di correzioni di dati già registrati. Queste richiedono però, da parte dell'utente, se si vuole mantenere un corretto collegamento tra gli archivi, una serie di passi non difficili ma forse fastidiosi.

E' consigliabile, quindi, controllare con grande scrupolo l'introduzione dei numeri delle estrazioni settimanali e correggerli, se errati, **prima** di dare il via libera per la registrazione e le contestuali modifiche dei molti archivi ad essi collegati. In tal modo ci si assicura l'acquisizione in forma semplice e continua delle molte informazioni elaborate.

```
Napoli   Ambi con numero 43 dal 02/01/88 al 05/01/91    :
Ai numeri che formano ambo con 43 segue n. settimana    :
:
 1 --     16 --     31 --     46 --     61 --     76 --
 2 --     17 --     32 --     47 135    62 --     77 --
 3 --     18 --     33 --     48 --     63 --     78 --
 4 --     19  80    34 --     49  80    64 --     79 143
 5 --     20 --     35  56    50 135    65 --     80  56
 6 --     21 --     36 --     51 135    66 --     81 --
 7  56    22 --     37 --     52 --     67 --     82 --
 8 --     23  80    38 135    53 --     68 --     83  80
 9 --     24 --     39 --     54 --     69 --     84 --
10 --     25 --     40 143    55 --     70 --     85 --
11 --     26 --     41 --     56 --     71 --     86 143
12 --     27 --     42 --     57 --     72 --     87 --
13 --     28 --     43 --     58 --     73  56    88 --
14 143    29 --     44 --     59 --     74 --     89 --
15 --     30 --     45 --     60 --     75 --     90 --
```

```
Numeri con piu' ritardi
totali al 05/01/91
Numeri    Rit.totali
 23    ->    299
 44    ->    291
 34    ->    284
 68    ->    283
 24    ->    256
 90    ->    254
 26    ->    251
 5     ->    247
 55    ->    241
 15    ->    238
 10    ->    237
 59    ->    231
 61    ->    230
 36    ->    229
 11    ->    227
```

Correzioni

L'ultima opzione consente la correzione di dati già registrati. Tutti possono essere modificati, utilizzando le scelte previste nel menu secondario delle correzioni.

Tale possibilità viene offerta, comunque, con alcune volute limitazioni, per evitare scompaginamenti, non sempre recuperabili, dei tanti archivi di uso.

```
Maggiori ritardi di AMBI su
tutte le Ruote in 158 sett.
RITARDI 157 :
 13 16*  37 42*
RITARDI 156 :
 15 66*  28 37*  44 85*
RITARDI 155 :
  9 15*   9 36*  18 80*
 36 77*  36 80*
RITARDI 154 :
  2 70*   8 14*  39 74*
RITARDI 153 :
 13 42*  41 73*
RITARDI 152 :
  5 51*
RITARDI 151 :
 15 57*  18 71*  24 88*
RITARDI 150 :
 69 72*
```

Una premessa che vale per tutte le operazioni consigliate in seguito: i ritardi, come è noto, sono calcolati, in ogni caso, valutando il numero di settimane che intercorrono tra l'ultima (in cui un numero o una qualsiasi combinazione di numeri sono usciti) e la settimana in corso.

Nel nostro caso, poiché ad ogni settimana corrisponde nell'archivio delle estrazioni (**Estraz.lot**) un numero che la individua, per definire la settimana cui corrisponde un particolare ritardo, è sufficiente semplicemente sottrarre, dal numero corrispondente all'ultima settimana registrata, quello del ritardo.

Un elenco totale delle settimane registrate (numeri e date) è invocabile con l'opzione n. 7 del menu principale (Ricerca delle estrazioni). Se, per esempio, al n. 55, sulla Ruota di Torino, corrisponde il ritardo 20 e l'ultima settimana registrata ha numero 150, il 55 (se i dati in archivio sono esatti!) si troverà sulla Ruota di Torino, nella settimana registrata con il n. 130 (150 - 20). Questo è tutto e, come per un numero singolo, vale per ogni combinazione di essi. Chiedendo quindi la settimana 130 con la specifica opzione (n. 7 del menu principale) se ne potrà controllare la validità.

```
Assenza su tutte le Ruote
Primo___numero 43
Secondo_numero 78
Ambo 43 78 assente
da 37 settimane  :
Ultima uscita nella sett. 121
```

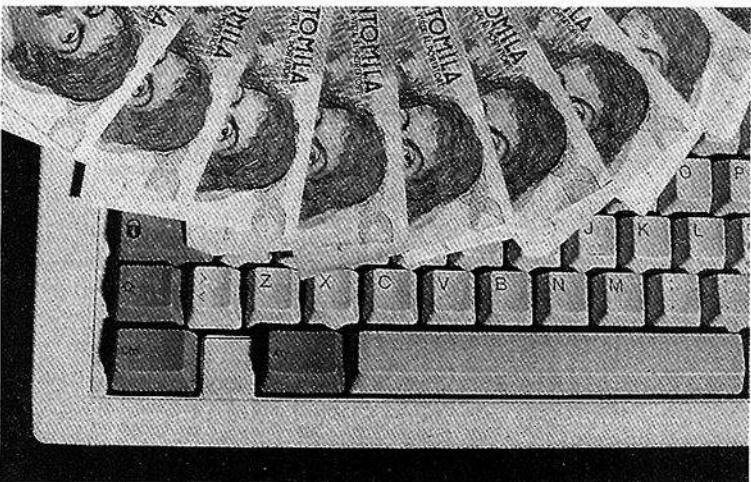
Controlli

Vediamo ora cosa succede se, consultando una rivista specializzata, si scopre che un qualsiasi ritardo non coincide con quello presente nel nostro archivio, dedicato a quel tipo di ritardo.

E' ovvio che, se l'informazione è esatta, nell'introduzione dei numeri relativi a due settimane registrate (quella corrispondente al ritardo indicato dalla rivista e quella corrispondente al "nostro" ritardo), si è verificato qualche errore. Poiché il computer non può sapere con quale altro numero è stato scambiato quello oggetto dell'indagine, occorre fornirsi di copie dei bollettini ufficiali delle estrazioni (presso le Ricevitorie del Lotto) e confrontare pazientemente i dati in essi presenti con quelli da noi registrati per le stesse settimane per individuare la fonte dell'errore.

Presa accurata nota dei numeri, da modificare in seguito nell'archivio delle Estrazioni, (oltre alla Ruota e alle date delle settimane), si può procedere a correggere il ritardo inesatto utilizzando la scelta apposita nel menu delle correzioni (la correzione riguarda solo l'archivio modificato e si svolge sotto gli occhi dell'utente). Conviene però sempre effettuare anche le correzioni appropriate delle estrazioni, relative alle settimane interessate, per due motivi: il primo per avere dati esatti nell'archivio delle Estrazioni; il secondo derivante dal fatto che, effettuate le correzioni, il programma riscrive anche l'archivio degli ambi e cambia le frequenze. Se l'ultimo fatto può non avere eccessivo interesse, l'altro non solo consente di adeguare i ritardi degli ambi (nelle varie possibilità di scelta) al contenuto attuale del file, ma offre anche la chance per ripristinare il collegamento di altri archivi di ritardi con quello corretto.

```
Estraz. 04/06/88  sett. 23

Bari        9  75  84  12  82
Cagliari   82  51  74  19  14
Firenze    27  72  30  29   7
Genova     71  38  13  21  64
Milano      3  82  28  76  83
Napoli     13  41  61  47  15
Palermo     5  53   3  76  31
Roma       86  68  15  31   8
Torino     24  32  18  35  67
Venezia    36   7  70  63  41
Tasti: a=avanti  z=indietro
p=stampa Altro tasto esce
```

Curiosando nell'archivio delle estrazioni, cosa notevolmente agevole e veloce, e in quella degli ambi, si possono infatti trovare indicazioni adeguate per correggere anche altri ritardi. Se per esempio si scopre che al posto del 55 si è digitato, per errore, il 48 (esempio precedente), si possono scorrere le estrazioni, all'indietro, a partire dall'ultima, ed esaminare i numeri della Ruota di Torino.

La prima settimana in cui tale numero sarà presente, fornirà la indicazione del ritardo maggiore del numero su detta Ruota (basta sottrarre, come già premesso, dal numero corrispondente all'ultima registrata quello della settimana trovata).

Questa stessa modalità si può utilizzare per altri ritardi, come anche per le eventuali combinazioni, nuove, che si creino (decine, cadenze, eccetera.), o che si perdano, in conseguenza di sostituzione di numeri al posto di altri preesistenti. Un suggerimento a questo punto per i numeri abbinati: prima di scorrere le estrazioni è opportuno verificare, con l'archivio degli ambi (dopo aver corretto ovviamente le estrazioni), se esistono ambi costituiti da quei numeri, sulla Ruota interessata, e regolarsi in conseguenza.

Per tutti questi motivi e non desiderando moltiplicare gli errori se, oltre a quello scoperto casualmente, ce ne fossero altri sempre dovuti a inesattezze nell'introduzione delle estrazioni settimanali, si è giudicato opportuno permettere di modificare i dati, ma limitatamente agli archivi interessati (a parte quelli degli Ambi, che vengono riscritti completamente se ci sono state correzioni delle estrazioni).

L'utente d'altra parte, con poche manovre, può estendere le correzioni se ha dati certi, e, impadronitosi dell'uso del programma, può escogitare, con un po' di fantasia, soluzioni adeguate al problema del momento, che non è sempre solubile allo stesso modo.

Si considerino, per esempio, i ritardi dei gemelli e dei vertibili, che sono i soli, relativi agli abbinamenti previsti, che vanno, a volte, ben oltre i dati contenuti nell'archivio delle Estrazioni (valori assoluti perfino di molti anni). In questi casi la ricerca degli Ambi completi permette (a correzioni avvenute) di riconoscere, per la Ruota interessata, se in archivio ci sono o meno altre evenienze della coppia modificata. Se non ce ne sono, occorre ripristinare i ritardi giusti con l'aiuto di qualche Rivista specializzata affidabile, così come ha fatto l'autore all'atto della impostazione iniziale del programma.

Si ricordi che, al momento dell'uscita di numeri e di coppie, i ritardi relativi si azzerano e comincia un nuovo ciclo, che si chiude, appena riescono gli stessi numeri e le stesse coppie sulla medesima ruota, con la qual cosa inizia un altro nuovo ciclo e così via.

In conclusione, è pur sempre vero che, nelle più varie situazioni, vale la considerazione che è meglio prevenire che curare.

Concludendo

Al termine del sintetico(!) esame delle operazioni realizzabili con il programma, appaiono manifeste le ampie possibilità offerte, anche per realizzare confronti finalizzati di dati.

Con il progressivo aumento delle settimane registrate, su cui fanno perno tutte le elaborazioni, alcune possibilità, tra cui quelle relative alla ricerca o al ritardo degli ambi e alla valutazione delle frequenze, saranno certamente esaltate. Ciò grazie alla sempre maggiore disponibilità di informazioni, che sul dischetto sono relative a ben 3 anni di estrazioni.

Tutte le altre possibilità si fondano, già ora, su valori assoluti dei dati (in parte ricavati da fonti affidabili) e sono, pertanto, esaurientemente soddisfatte. Sono tra esse comprese quelle più comunemente desiderate: ritardi dei numeri, delle decine, delle cadenze, dei gemelli, dei vertibili.

E per Amiga?

Il programma, come già detto, è stato sviluppato in C e non presenta particolari istruzioni legate all'hardware della macchina. Abbiamo affidato, ad un nostro esperto collaboratore, l'adattamento del programma Ms - Dos all'Amiga e, al momento in cui leggerete queste note, il pacchetto sarà disponibile anche per questo computer.

Si consiglia, tuttavia, di telefonare in Redazione per verificare l'effettiva disponibilità del dischetto, prima di ordinarlo per corrispondenza.

Archivi

Ricordiamo ancora una volta che il dischetto contiene un archivio di ben tre anni di estrazioni e sarà periodicamente aggiornato. Chi desidera entrare in contatto con l'autore del programma, può scrivere in Redazione specificando le richieste o i suggerimenti da apportare per la prossima edizione.

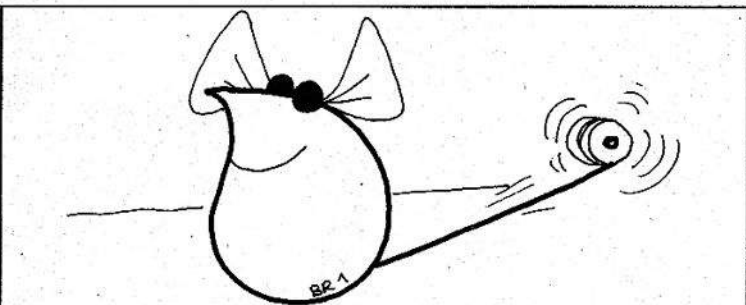
Se, infatti, il prodotto avrà successo, provvederemo a divulgare versioni sempre più aggiornate e potenti.

Come procurarsi il disco

Il dischetto **LOTTO Saggese by Systems** non viene posto in vendita in edicola a causa del limitato interesse, in campo informatico, di programmi di tale tipologia.

E' quindi possibile ordinarlo per corrispondenza, al nostro servizio arretrati, al prezzo di L. 30000 (trentamila), comprese le spese di spedizione. Ricordiamo che non ci è possibile inviare materiale contrassegno e che è assolutamente indispensabile indicare, al momento dell'ordinazione, il formato desiderato del dischetto (5.25 oppure 3.5 pollici).

Al momento di andare in stampa non possiamo garantire la disponibilità del formato per Amiga (telefonare per conferma). A causa della complessità della procedura, precisiamo che il pacchetto NON è disponibile (nè mai lo sarà) per il C/64 - 128 o altri piccoli ed obsoleti computer Commodore.

Terminiamo precisando che la procedura gira su qualsiasi Ms - Dos compatibile (di qualsiasi marca) anche nella configurazione minima e con monitor in bianco e nero. Il listato sorgente è presente nello stesso dischetto e caricabile direttamente in ambiente Turbo C Borland.

UN... "GENERATORE" DI COMPITI IN CLASSE

Quando le classi sono numerose, ed un "verifica" rapida si rende necessaria, un computer può essere di aiuto per l'insegnante.

Le disposizioni ministeriali sulla formazione delle classi, che impongono un numero massimo di alunni per ciascuna di esse, vengono tradizionalmente disattese dagli organi *competenti* (si fa per dire); anzi, maggiori sono le restrizioni annunciate, maggiore sarà il numero di alunni stipati in aule, possibilmente anguste e buie.

Polemiche a parte, ogni insegnante giunge, prima o poi, al momento cruciale delle **verifiche** (che un tempo si chiamavano semplicemente interrogazioni o compiti in classe). Le interrogazioni, se svolte seriamente, richiedono tempi non modesti. I compiti in classe, invece, presentano due problemi di un certo rilievo.

Il primo è rappresentato dal tempo richiesto per la correzione, soprattutto per quelle discipline per le quali un qualsiasi tentativo di automatizzare la procedura di correzione cade miseramente (tutta la nostra solidarietà agli insegnanti di lettere). In questo caso, pur se la correzione dei compiti fa parte del lavoro di un docente, non si può negare che la prospettiva di dedicarvi molte ore, in ambito extrascolastico, può scoraggiare un frequente ricorso a verifiche scritte.

Un po' meglio vanno le cose per gli insegnanti di matematica che, imponendosi un metodo simile ad una catena di montaggio, possono sveltire la procedura; a patto di assegnare un'unica "traccia".

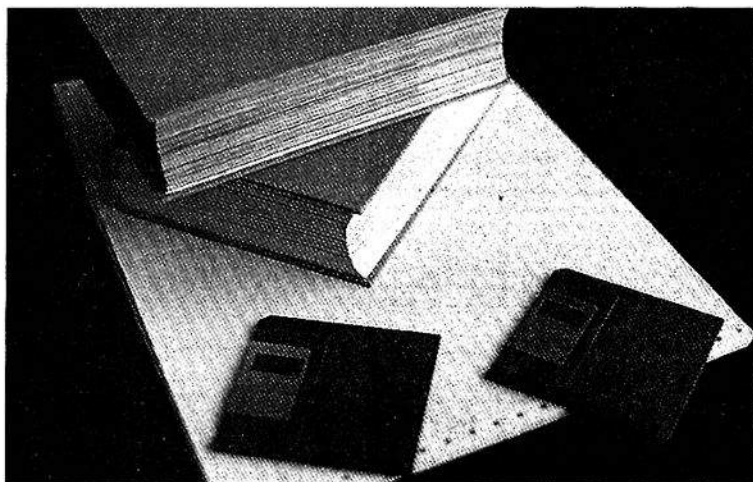
Assegnando un tema di italiano capita di rado che un allievo si azzardi a copiare il compito dal compagno di banco, per ovvi motivi.

Nelle prove scritte di matematica, al contrario, due compiti eguali, per non dire identici, non costituiscono, giuridicamente, prova di copia. Da che mondo è mondo, infatti, tutti gli studenti praticano senza vergogna (vergognarsi di che cosa, del resto?) l'esercizio della copia in classe; molti sono i contusi che tentano di sottrarsi a simili pratiche; numerosi, del resto, sono gli amori sbocciati tra compagni di sesso opposto, amori spesso iniziati a causa un'esagerata manifestazione di riconoscenza.

Il problema si potrebbe risolvere assegnando, a ciascun allievo, un compito diverso; in effetti alcuni docenti, pur non arrivando a tanto, entrano in classe con

```
callegari luigi
de_simone alessandro
maggi michele
miotti marco
```

Esempio di file-alunni

I nomi degli allievi vanno memorizzati in un file ASCII digitandoli in minuscolo; al momento opportuno sarà necessario indicare l'intero path di ricerca.

```
' Generatore di compiti in classe
' Schema generale + esempio
DECLARE SUB Esercizio ()
DECLARE SUB Stampa (X, separa$)
DECLARE SUB Casuale (X)
DECLARE SUB Stringa (X, X$)
DECLARE SUB Stampante (X$)
DECLARE SUB LeggiAlunni (X$, X)
DECLARE SUB AssegnaNomi (X$, X)
DECLARE SUB AttendiTasto (X$)
DIM SHARED A$(6), Stamp$, NC, Acapo$, Data$, Nfile$
REM Inizio prima parte specifica
FOR I = 1 TO 6: READ A$(I): NEXT
DATA "Un parallelepipedo misura (a) cm. "      : REM 1
DATA "; (b) dm. 0" , "; (h) m. 0"             : REM 2, 3
DATA "; pesa (Pp) g. "                         : REM 4
DATA "Quale  il suo volume V (cm^3)? "           : REM 5
DATA "Quale  il suo peso specifico Ps (Kg/m3)? ": REM 6
REM Fine prima parte specifica
CLS : tipo$ = "": INPUT "Compito, Esercizio singolo
(c/e)"; tipo$
IF UCASE$(tipo$) = "E" THEN CALL Esercizio: END
CLS : INPUT "File Alunni (es.  b:\nomi\alunni) "; Nfile$
PRINT "Leggo nomi dal file: "; Nfile$
CALL AttendiTasto(X$)
IF X$  CHR$(13) THEN END
CALL LeggiAlunni(X$, X): NC = X: CALL AttendiTasto(X$)
IF X$  CHR$(13) THEN END
DIM SHARED al$(NC)
CALL AssegnaNomi(X$, X)
Stamp$ = "": INPUT "Video / Stampante"; Stamp$
IF UCASE$(Stamp$) = "S" THEN
Stamp$ = "printer": INPUT "Data compito"; Data$
END IF
IF Data$ = "" THEN Data$ = DATE$
ls = 79: IF Stamp$ = "printer" THEN ls = 130
FOR I = 1 TO ls: separa$ = separa$ + "-": NEXT
DIM SHARED La(NC), Lb(NC), Lc(NC), Pg(NC), Volume(NC),
Ps(NC)
FOR I = 1 TO NC
REM Inizio seconda parte specifica
Ps(I) = 0: WHILE Ps(I) = 0
X = 0: WHILE X  5: X = 10
CALL Casuale(X): WEND: La(I) = X          ' la (cm.)
X = 0: WHILE X  1: X = 10
CALL Casuale(X): WEND: Lb(I) = X / 10     ' lb (dm.)
X = 0: WHILE X  30: X = 50
CALL Casuale(X): WEND: Lc(I) = X / 100 ' h ( m.)
Volume(I) = La(I) * (Lb(I) * 10) * (Lc(I) * 100)
Pg(I) = INT(((La(I) * Lb(I) * 10 * Lc(I) * 100) * .5) /
10) * 10     ' pg. peso (g.)
Ps(I) = Pg(I) / (La(I) * Lb(I) * Lc(I))
' ps() peso specifico (kg./m3)
WEND
REM Fine seconda parte specifica
NEXT I: CALL Stampa(NC, separa$)
END
```

tre o quattro tracce diverse, che vengono assegnate, alternativamente, una per ciascuna fila di banchi. In questi casi gli studenti si sottopongono a pratiche umilianti: colli che si allungano esageratamente nel tentativo di superare due file di banchi; bigliettini microscopici da ingoiare, senz'acqua, in caso di controllo; segnali visivi ai limiti della decifrabilità; colpi di tosse di tipo comatoso per richiamare l'attenzione dei compagni, ma non dell'insegnante (che non è propriamente un fesso, ma fa finta di non sentire e non vedere, impietosito dalle gocce di sudore freddo che scorrono sui fogli dei suoi alunni).

Una percentuale notevole di tempo ed energia, insomma, viene dissipata dagli allievi per escogitare sistemi di comunicazione che risultino economicamente più vantaggiosi del dispiego di energie necessarie per studiare con calma e prepararsi adeguatamente per il compito.

Se, però, ogni allievo avesse la matematica certezza di dover svolgere un compito che in nessun caso potrà risultare eguale a quello assegnato ad un compagno, si assisterebbe a un duplice fenomeno (volendo escludere l'arruolamento nella legione straniera): durante le ore assegnate per lo svolgimento, ognuno lavorerebbe per proprio conto; il corretto svolgimento di un compito, inoltre, darebbe sufficienti garanzie di paternità responsabile.

"Inventare" 30 compiti diversi, però, è sufficientemente allucinante da scoraggiare una simile procedura (ci pensate al tempo necessario per la correzione?).

Se, però, è un computer a generare dati diversi, e a correggerli in seguito, la faccenda può divenire interessante.

Automatismi

E' necessario precisare che, questa pubblicata, è solo la versione di una procedura che può diventare più sosfisticata ed automatizzata.

Non è infatti possibile pretendere la stesura di un programma che, in modo del tutto autonomo, riesca (anche se opportunamente guidato) a "pensare" ad un problema, ed alla sua soluzione.

Il linguaggio usato è uno dei più semplici da comprendere (cioè il **QuickBasic**

```
SUB AssegnaNomi (X$, X)
REM Esamina il file contenenete i nomi e trasforma
REM i primi caratteri (nome e cognome) in maiuscolo.
X = 1: OPEN Nfile$ FOR INPUT AS 2
WHILE EOF(2) = 0: INPUT #2, al$(X): l = LEN(al$(X))
z$ = "": g = 0
FOR I = 1 TO l: w$ = MID$(al$(X), I, 1): y$ = UCASE$(w$)
IF g  0 THEN g = g + 1
IF w$ = " " THEN g = g + 1
IF g  2 THEN y$ = LCASE$(w$)
z$ = z$ + y$: NEXT: al$(X) = z$: X = X + 1
WEND: CLOSE 2
END SUB

SUB AttendiTasto (X$)
REM Attende la pressione di Return
X$ = ""
PRINT : PRINT "Premi Return per continuare.";
PRINT " (altro tasto per interrompere)"
WHILE X$ = "": X$ = INKEY$: WEND
END SUB

SUB Casuale (X) STATIC
REM Genera un numero casuale
RANDOMIZE (TIMER): X = INT(RND(1) * X)
END SUB

SUB Esercizio
REM Consente di svolgere un esercizio alla volta
CLS : X$ = CHR$(13)
WHILE X$ = CHR$(13): PRINT
REM Inizio terza parte specifica
PRINT A$(1); : INPUT La: REM misura lato A (cm.)
PRINT A$(2); : INPUT Lb: REM Misura lato B (dm.)
PRINT A$(3); : INPUT Lc: REM Altezza (m.)
PRINT A$(4); : INPUT Pg: REM Peso (g.)
Volume = La * (Lb * 10) * (Lc * 100)
PRINT "Volume: "; Volume; "(cm^3)"
Ps = Pg / Volume
PRINT "Peso specifico= "; Ps * 1000; "(Kg./m^3)"
REM Fine terza parte specifica
PRINT : X$ = "": CALL AttendiTasto(X$): PRINT
WEND
END SUB
SUB LeggiAlunni (X$, X)
REM Determina il numero di allievi
REM presenti nel file, allo scopo di
REM dimensionare correttamente il vettore
REM corrispondente: DIM AL$(nc)
OPEN Nfile$ FOR INPUT AS 2
WHILE EOF(2) = 0
INPUT #2, X$: X = X + 1
PRINT X$
WEND: CLOSE 2
END SUB
SUB Stampa (X, separa$)
IF Stamp$ = "printer" THEN SHELL "mode lpt1:132"
```

Microsoft) facilmente traducibile in **AmigaBasic** oppure in **Pascal**. Dal momento che, però, non è necessaria una elevata velocità di elaborazione, che cozzerebbe contro la lentezza intrinseca della stampante, ci si può limitare al popolare interprete - compilatore.

La procedura è strutturata in varie parti, ognuna individuabile da una subroutine, alcune delle quali di tipo "universali", altre da modificare a seconda dei casi.

Nel programma pubblicato è "incorporato" un esempio, volutamente banale per non appesantire il lavoro di digitazione da parte dei nostri lettori, che si presta bene a personalizzazioni di vario tipo.

Che cosa fa il programma

Il programma genera tanti compiti, rigorosamente diversi l'uno dall'altro, quanti sono gli allievi iscritti in una classe. I nomi di questi devono esser presenti in un file sotto forma di un comunissimo file ASCII, precedentemente memorizzato su un dischetto.

Ricordiamo che, nel riquadro specifico, è presente un esempio di file contenente i nomi degli allievi (scritti usando esclusivamente caratteri **minuscoli**); tale file potrà, ovviamente, essere usato anche da altri programmi, da sviluppare in seguito, che dovessero riferirsi ai componenti di una classe.

Il programma fornisce le misure dei tre lati di un parallelepipedo (rispettivamente in: centimetri, decimetri, metri) ed il suo peso in grammi. Viene quindi richiesta la misura del volume (in centimetri cubici) ed il suo peso specifico (in chilogrammi a metro cubo).

Nonostante la banalità del compito, questo potrà rivelarsi già utile, almeno nella scuola dell'obbligo, dopo una lezione sulle **equivalenze** (da centimetro a metro, da grammi a chilogrammi, da centimetro cubo a metro cubo) e sulla determinazione del **peso specifico**.

Quando gli allievi consegneranno il compito, l'insegnante, con estrema rapidità, sarà in grado di stabilire, confrontando le risposte fornite dagli allievi con quelle presenti sul tabulato, se la lezione è stata recepita, oppure se necessita una qualche precisazione.

Inutile dire che la correzione "automatica" non tiene conto del procedimento seguito dagli allievi (spesso un singolo errore commesso all'inizio pregiudica il

```
IF Stamp$ = "printer" THEN OPEN "LPT1:" FOR OUTPUT AS
#1: WIDTH #1, 132
FOR k = 1 TO X
X$ = "(" + Data$ + ") COMPITO N." + STR$(k) + " " +
al$(k): Acapo$ = "s": CALL Stampante(X$)
REM Inizio quarta parte specifica
X$ = A$(1) + STR$(La(k)): Acapo$ = "n"
CALL Stampante(X$): REM Un parall...misura cm.
CALL Stringa(Lb(k), X$): X$ = A$(2) + X$: Acapo$ = "n":
CALL Stampante(X$): REM dm. ..
CALL Stringa(Lc(k), X$): X$ = A$(3) + X$: Acapo$ = "n":
CALL Stampante(X$): REM e m. ...
CALL Stringa(Pg(k), X$): X$ = A$(4) + X$: Acapo$ = "s":
CALL Stampante(X$): REM pesa...
X$ = A$(5): Acapo$ = "n"
CALL Stampante(X$): REM Quale  il volume?
X$ = A$(6): Acapo$ = "s"
CALL Stampante(X$): REM Quale  il peso specifico?
X$ = separa$: Acapo$ = "s"
CALL Stampante(X$): REM spazio tra un compito e l'altro
REM Fine quarta parte specifica
NEXT
X$ = " ": Acapo$ = "s": CALL Stampante(X$)
X$ = "(Compito del " + Data$ + ")": Acapo$ = "s"
CALL Stampante(X$)
X$ = " ": Acapo$ = "s": CALL Stampante(X$)
FOR k = 1 TO X
X$ = "COMPITO N." + STR$(k) + " " + al$(k) + " "
Acapo$ = "n": CALL Stampante(X$)
REM Inizio quinta parte specifica
X$ = "Volume = " + STR$(Volume(k)): Acapo$ = "n"
CALL Stampante(X$)
X$ = ": ps =" + STR$(Ps(k)): Acapo$ = "n"
CALL Stampante(X$)
X$ = separa$: Acapo$ = "s": CALL Stampante(X$)
REM Fine quinta parte specifica
NEXT: CLOSE 1
END SUB

SUB Stampante (X$) STATIC
REM Invia i messaggi al video o alla stampante
REM a seconda della selezione effettuata
IF Stamp$ = "printer" THEN
   PRINT #1, X$;
      IF Acapo$ = "s" THEN
        PRINT #1,
      END IF:
END IF
  PRINT X$; : IF Acapo$ = "s" THEN PRINT
END SUB

SUB Stringa (X, X$) STATIC
REM Esclude il carattere a sinistra di un valore numerico
REM allo scopo di stampare, ad esempio.. 0.36 e non 0. 36
X$ = RIGHT$(STR$(X), LEN(STR$(X)) - 1)
END SUB
```

risultato finale, a dispetto di un corretto procedimento). Tuttavia l'elevata velocità di controllo, permessa dalla procedura computerizzata, può essere di aiuto in vari casi.

Come gira il programma

La prima domanda che compare, non appena il programma parte, si riferisce alla scelta tra un **compito in classe** ed un **esercizio singolo.**

La seconda opzione, che invitiamo a sperimentare per prima, ha due scopi. Anzitutto consente di verificare la corretta trascrizione del programma, dal momento che l'algoritmo per il calcolo dei valori è sempre lo stesso. Rispondendo, quindi, **"E"** (e premendo Return) verrà chiesta la misura del primo lato del parallelepipedo (in centimetri). La seconda domanda si riferisce alla misura del secondo lato (da esprimere in decimetri) e la terza all'altezza (in metri).

In effetti, prima della domanda, compaiono altri caratteri apparentemente strani. Non sono altro che le stringhe che verranno trascritte su carta, in seguito, al momento della stampa e che, in quella sede, avranno la loro importanza. Per non complicare le cose, infatti, tale apparente "inestetismo" non è stato eliminato su video, anche perchè l'opzione "E" risulta di *uso interno*, vale a dire che verrà usata solo dall'insegnante.

Il secondo motivo che giustifica la presenza dell'opzione, risiede nella possibilità di determinare la soluzione di un qualsiasi problema che richieda, in ingresso, gli stessi dati.

La possibilità di eseguire singoli esercizi è attiva finchè si preme, dopo l'apparire di ciascuna soluzione, il tasto Return. Premendo un altro tasto, infatti, il programma si interrompe.

Facendolo *ripartire*, e premendo stavolta **"C"** alla domanda, verrà attivata la parte del listato che genera, propriamente, un compito in classe.

Dal momento che si è voluto rendere la procedura quanto più automatica possibile, risulta indispensabile indicare il path (percorso) preciso da seguire per individuare il file contenente i nomi degli allievi. Supponendo che questo lo abbiate memorizzato in...

a:\nomi\prima.a

...e che si riferisca, ad esempio, alla classe "prima A", digiterete esattamente

quanto indicato prima. Del resto, in caso di errore, il programma si interrompe emettendo il messaggio *"File not found"*, senz'altra conseguenza.

Un messaggio di o.k. compare, invece, in caso affermativo. Per ulteriore conferma i nomi degli allievi verranno visualizzati dopo la pressione del tasto return, in modo da interrompere la procedura nel caso si sia indicata, per errore, una classe diversa.

Un'altra domanda, relativa alla scelta tra video e stampante, viene formulata. Questa, in effetti, servirà per evitare spreco di carta dal momento che consente di verificare, su video, la corretta trascrizione del programma. In seguito potrà essere evitata.

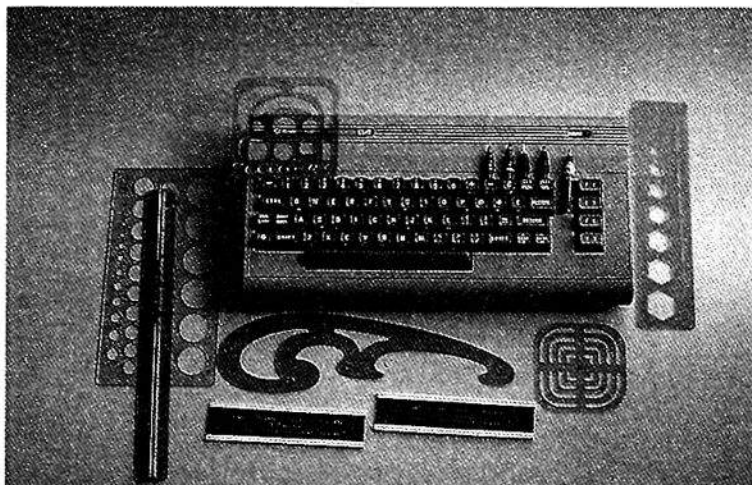
Rispondendo **"V"** compariranno, su video, tutte le tracce dei compiti. In coda seguiranno le rispettive soluzioni. Dal momento che la visualizzazione non si interrompe, si consiglia di creare, all'inizio, un file fittizio contenente solo **due** nomi di ipotetici allievi: la "redirezione" su video, infatti, serve solo per controllare la correttezza della trascrizione del listato e l'allineamento dei messaggi.

Se, invece, premete **S**(tampante) alla domanda sull'output, il programma chiederà la data che si desidera far comparire sui compiti (premendo Return "a vuoto" verrà usata la data del sistema in quel momento attiva) e poi riverserà su carta i singoli compiti e le soluzioni. Un micro esempio dell'output è riportato in queste stesse pagine.

Come funziona

Alcune subroutine sono di utilizzo generale e possono essere usate senza apportare alcuna modifica in caso di personalizzazione. Incominciamo, appunto, ad esaminarle una per una.

Sub Stringa (X, X$)

Trasforma il valore associato a **X** nella stringa **X$**; di questa, poi, considera tutti i caratteri ad eccezione del primo (il segno o lo spazio bianco). Serve per un corretto allineamento dell'eventuale valore decimale. Molti allievi, infatti, non sono abituati ad interpretare correttamente, ad esempio, il valore "zero virgola settantacinque" espresso come (.75), ma solo se indicato come 0.75 (ed è già molto se si convincono ad usare il *punto* invece della *virgola*...).

La subroutine non si presta, ovviamente, per la visualizzazione di valori negativi che, in tal caso, verrebbero privati del segno!

Sub Stampante (X$)

A seconda del contenuto della stringa **Stamp$**, invia al video (o alla stampante) la stringa X$ (notare il punto e virgola successivo, che consente di far proseguire la stampa sullo stesso rigo).

Se, poi, la variabile **Acapo$** è **"S"**, viene inviato un **Print#** a vuoto per interrompere l'allineamento.

In pratica la Sub rappresenta una rudimentale forma di formattazione del testo e viene usata intensivamente nella subroutine Stampa (X, Separa$).

Sub LeggiAlunni (X$, X)

Cerca il file che dovrebbe contenere i nomi degli allievi e, se lo trova, ne determina il numero. Serve per dimensionare automaticamente i vettori destinati a contenere i vari dati per ciascun allievo.

Sub Casuale (X)

Genera un numero casuale il cui valore massimo è determinato dal valore in ingresso di X.

Sub AttendiTasto (X$)

Interrompe l'elaborazione finchè non viene premuto un tasto qualsiasi. Questo, al ritorno dalla sub, può essere esaminato per determinare il da farsi.

Sub AssegnaNomi (X$, X)

Rilegge il file che contiene i nomi degli allievi e li pone, uno per uno, nel vettore **AL$()**. Trasforma in maiuscolo **tutti** i caratteri del cognome ed il **primo** del nome. Nel caso di cognomi formati da due parole (de simone, la sorella, ecc.) queste devono esser digitate separandole dal meno shiftato (de_simone, la_sorella).

❑❒

Le routine specifiche

Le altre routine non possono essere automatizzate, ma richiedono un lavoro di adattamento da parte del lettore. Vediamo, quindi, quali sono e come sono strutturate, ricordando che sono individuabili, ciascuna, da due Rem relative, rispettivamente, all'inzio ed alla fine della parte da "personalizzare".

Programma

Il programma principale inizia, tradizionalmente, con una serie di **Declare Sub** relative a tutte le sub che si incontreranno.

I **Data** presenti devono esser tanti quante sono le frasi che si intende far apparire nel compito (nel nostro caso, **6**). Un banale ciclo For... Next le "incamera" nel vettore A$(). Seguono, quindi, tutti i comandi che possono esser considerati "universali" e che verranno lasciati tal quali in altre, eventuali personalizzazioni. Le ultime righe del programma principale (racchiuse dall'altro ciclo For... Next) attivano la ricerca casuale dei dati.

Si noti come la determinazione dei quattro dati (lato a, lato b, altezza, peso) sia possibile "guidarla" assegnando un valore massimo ad X ed imponendo la ripetizione della ricerca casuale nel caso

in cui il valore determinato sia minore (esempio: If X < 5) di un minimo stabilito a priori. Tale accorgimento consente di controllare il grado di omogeneità dei compiti: tutti diversi, ma tutti simili...

Il volume viene automaticamente assegnato ai singoli elementi del vettore **Volume()**; per l'assegnazione del peso, invece di ricorrere alla ricerca di un numero casuale, si è preferito usare una formuletta che, grossomodo, rendesse credibile il dato; questo, infatti, è inferiore all'unità e corrisponde ad una gran varietà di legni esistenti. Nulla vieta di generare valori compatibili anche con pesi specifici maggiori (come quelli dei metalli) o del tutto fantasiosi. L'ultima formula che compare nel programma principale (Ps(i)=...) determina con precisione il peso specifico del parallelepipedo.

Sub Stampa (X, Separa$)

Questa routine si incarica di riportare su carta i risultati dell'elaborazione. Sfrutta adeguatamente i 6 (nel nostro caso specifico) messaggi prima memorizzati in A$(i) e ricorre ad altre sub della procedura.

Anzitutto, se è stata selezionata la stampante, questa viene attivata in modalità **132 colonne** per consentire un migliore allineamento.

In seguito viene "costruita" la stringa **X$** ed inizializzata la stringa **Acapo$** (con "**s**" oppure "**n**") prima di invocare la Sub Stampante (X$). Il ciclo viene ripetuto (For... Next) per tutti i nomi degli allievi presenti nel vettore AL$(). Successivamente, un secondo ciclo For... Next riporterà, ancora su carta, i risultati corrispondenti.

Personalizzazioni

Seguendo la falsariga dell'esempio pubblicato sarà possibile realizzare, anzitutto, un file-programma di uso generale che consenta, in futuro, di realizzare altri compiti "automatici". Con un po' di pazienza, infatti, la procedura si presta per inserire altre domande, cui corrisponderanno altrettante formule e soluzioni, che possono rendere il compito sempre più diversificato e impegnativo. Il programma, inoltre, si presta per assegnare i famigerati "compiti per le vacanze": il docente potrà regalare ai suoi allievi centinaia di compiti, e relative soluzioni, per tenerli impegnati nei mesi estivi. Per questo motivo l'articolo viene pubblicato anonimo; le rappresaglie, con i tempi che corrono, è bene evitarle *in fieri*...

```
(27 / 3 / 1991) COMPITO N. 1 CALLEGARI Luigi
Un parallelepipedo misura (a) cm. 8; (b) dm. 0.2; (h) m. 0.45; pesa (Pg) g. 350
Quale è il suo volume V (cm.^3)? Quale è il suo peso specifico Ps (Kg/m3)?


------------------------------------------------------------------------------------------
(27 / 3 / 1991) COMPITO N. 1 DE_SIMONE Alessandro
Un parallelepipedo misura (a) cm. 8; (b) dm. 0.2; (h) m. 0.45; pesa (Pg) g. 350
Quale è il suo volume V (cm.^3)? Quale è il suo peso specifico Ps (Kg/m3)?


------------------------------------------------------------------------------------------
(27 / 3 / 1991) COMPITO N. 3 MAGGI Michele
Un parallelepipedo misura (a) cm. 8; (b) dm. 0.2; (h) m. 0.45; pesa (Pg) g. 400
Quale è il suo volume V (cm.^3)? Quale è il suo peso specifico Ps (Kg/m3)?


------------------------------------------------------------------------------------------
(27 / 3 / 1991) COMPITO N. 4 Miotti Marco
Un parallelepipedo misura (a) cm. 9; (b) dm. 0.5; (h) m. 0.38; pesa (Pg) g. 850
Quale è il suo volume V (cm.^3)? Quale è il suo peso specifico Ps (Kg/m3)?


------------------------------------------------------------------------------------------
(Compito del 27 / 3 / 1991)
Compito N. 1 CALLEGARI Luigi         Volume = 720;         Ps = 486.1111
------------------------------------------------------------------------------------------
Compito N. 2 DE_SIMONE Alessandro    Volume  = 777.0001;   Ps = 489.0605
------------------------------------------------------------------------------------------
Compito N. 3 MAGGI Michele           Volume  = 810;        Ps = 493.8272
------------------------------------------------------------------------------------------
Compito N. 4 Miotti Marco            Volume = 1710;        Ps  = 497.076
```

Esempio di output su stampante

Si noti la completa leggibilità delle tracce dei compiti e la facilità con cui è possibile verificare le risposte fornite. Pur se il programma non pretende di sostituire l'insegnante, non c'è dubbio che una "collezione" di generatori di compiti in classe può aiutare il docente nel determinare il grado di apprendimento conseguito dagli allievi.

Una sfida al mese

TUTTE LE SILLABE DI QUESTO MONDO

Questo mese vi proponiamo lo sviluppo di un'applicazione abbastanza semplice, quasi banale. Tra il dire e il fare, però...

Uno dei più importanti algoritmi presenti in word processor di un certo livello è rappresentato dalla possibilità di dividere automaticamente in sillabe le parole che, stampate in prossimità della fine del rigo, siano troppo lunghe per essere ospitate per intero sul rigo stesso.

I w/p meno sofisticati risolvono il problema in modo piuttosto brutale, limitandosi a trascinare l'intera parola al rigo successivo inserendo, magari, spazi bianchi tra una parola e l'altra per consentire il corretto allineamento anche a destra (e realizzando la cosiddetta **giustificazione**).

Un w/p progettato in quest'ultimo modo si presta per essere usato in qualsiasi paese del mondo, dal momento che non è necessario implementare le regole della divisione in sillabe, che cambiano a seconda della lingua adoperata.

I pacchetti applicativi che consentono la divisione in sillabe, infatti, devono offrire anche la possibilità di selezionare la lingua; in quest'ultimo caso, poi, l'algoritmo deve funzionare in modo eccellente, senza eccezioni. Purtroppo (i lettori più attenti se ne saranno certamente accorti) il sistema di impaginazione da noi adoperato, per stampare la nostra rivista, a volte fa cilecca; lo dimostra il modo improprio di dividere in sillabe alcune parole, che non sempre sono le stesse, nè è possibile stabilire a priori il momento del bizzarro comportamento. Mah! La nostra speranza è che i lettori non attribuiscano a noi la causa della scorretta divisione...

L'algoritmo

Il programma di divisione in sillabe è già bello e pronto, ed è ripreso pari pari da un precedente fascicolo di Computer Club.

La sintassi, **Basic**, è di tipo rigorosamente universale (C/64, Vic 20, C/16, Plus/4, AmigaBasic, Gw - Basic, QuickBasic, Turbo Basic); la sua brevità, inoltre, si presta benissimo per scrivere l'implementazione in **Pascal** (Turbo oppure Quick) e in "**C**".

Si tratta di digitare la parola di cui si desidera la divisione in sillabe; il computer, dopo la pressione del tasto Return, visualizzerà queste ultime separandole con un trattino.

Per interrompere, basta rispondere zero (**0**) alla domanda. Tutto qua.

La sfida

La nostra proposta è quella di scrivere un programma, in qualsiasi linguaggio e per qualsiasi computer (possibilmente **Ms - Dos** oppure **Amiga**) che, tenendo conto di alcuni parametri forniti in ingresso, visualizzi (su video o stampante) un qualsiasi file ASCII, precedentemente indicato, incolonnato correttamente con le parole di destra eventualmente divise in sillabe.

In Input bisognerà indicare:

1) Il nome del file ASCII (ovvio) completo di eventuale path (percorso) per raggiungerlo.
2) L'output desiderato (video / stampante).
3) Margine sinistro e destro (espressi in numero di caratteri).
4) Giustificazione (sì, no).
5) Salto pagina, o pausa, dopo la stampa di un certo numero di righe (da indicare).

La procedura, come già detto, dovrà essere applicata a puri e semplici files ASCII, scritti mediante **text editor** e non ricorrendo a **word processor**; in quest'ultimo caso, infatti, l'immancabile presenza di caratteri di controllo renderebbe ardua l'estrazione delle parole e la conseguente elaborazione.

Non si tratta, quindi, di scrivere un word processor (ci mancherebbe altro!) ma semplicemente di implementare una procedura che tenga conto di alcuni problemi che si presentano al momento della formattazione di un testo.

Come e perchè

Il programma dovrà essere il più breve possibile; verrà infatti privilegiata la pubblicazione del listato che, a parità di parametri, risulti il più efficiente e semplice da capire (ma, soprattutto, da digitare...).

In ogni caso, si ricorda che vengono accettate solo proposte che perverranno tramite la nostra BBS (tel. **02 / 57.60.52.11**) oppure su disco, purchè preventivamente concordate per telefono (tel. **02 / 57.60.63.10**).

All'autore del lavoro giudicato migliore verrà, come di consueto, assegnato un congruo premio.

Sfida del mese

```
10 REM Divisone in sillabe  (versione universale)
20 s = 1
30 PRINT : INPUT "Parola  (0=fine)"; a$: PRINT
33 IF a$ = "0" THEN END
35 IF s > LEN (a$) THEN GOTO 20
40 x$ = MID$ (a$, s, 1): GOSUB 300
50 IF a = 0 THEN GOTO 250
70 x$ = MID$ (a$, s + 1, 1): GOSUB 300
80 IF a = 0 THEN GOTO 150
90 IF x$ = "i" THEN GOTO 120
100 IF MID$ (a$, s, 1) = "i" OR MID$ (a$, s, 1) = "u" THEN GOTO 250
110 GOTO 260
120 x$ = MID$ (a$, s + 2, 1): GOSUB 300
125 IF s > 1 THEN IF MID$ (a$, s - 1, 2) = "qu" AND a = -1 THEN PRINT MID$ (a$, s, 2); :
s = s + 2: GOTO 35
130 IF a = -1 THEN GOTO 260
140 GOTO 250
150 IF s + 2 > LEN (a$) THEN GOTO 270
160 x$ = MID$ (a$, s + 2, 1): GOSUB 300
170 IF a = -1 THEN GOTO 260
180 IF MID$ (a$, s + 1, 1) = MID$ (a$, s + 2, 1) THEN GOTO 270
190 a1$ = MID$ (a$, s + 1, 1): IF a1$ = "s" OR a1$ = "g" THEN GOTO 260
200 a2$ = MID$ (a$, s + 2, 1): IF a2$ = "r" OR a2$ = "l" OR a2$ = "h" THEN GOTO 260
210 GOTO 270
250 PRINT MID$ (a$, s, 1); : s = s + 1: GOTO 35
260 PRINT MID$ (a$, s, 1); "-"; : s = s + 1: GOTO 35
270 PRINT MID$ (a$, s, 2); "-"; : s = s + 2: GOTO 35
300 a = x$ = "a" OR x$ = "e" OR x$ = "i" OR x$ = "o" OR x$ = "u": RETURN
310 END
```

di Carlo d'Ippolito

DELUXE PAINT AMIGA + MS - DOS

Deluxe Paint è nato originariamente per Amiga ma "trapiantato", in seguito, nel mondo Ms - Dos

L'autore del programma grafico, **Dan Silva**, lo produsse per i primissimi modelli di **Amiga 1000**, in pratica per il lancio della macchina sul grande mercato americano.

Non sappiamo quanto abbia influito la disponibilità di un pacchetto come Deluxe Paint sulle vendite di Amiga 1000, ma è certo che è stato il programma di maggiore successo: ai tempi si calcolò che il 70% dei possessori statunitensi di Amiga 1000 avevano acquistato anche il pacchetto originale.

La versione **Ms - Dos** di Deluxe Paint è molto simile a quella per Amiga. Mentre, per Amiga, Deluxe Paint è uno tra *alcuni* programmi di grafica (Express Paint, Photon Paint, Digipaint), molti dei quali equivalenti come prestazioni, nel mondo Ms - Dos è probabilmente *il più semplice e potente*, allo stesso tempo, pacchetto di grafica creativa bidimensionale.

Vogliamo riportare alcuni cenni d'uso del pacchetto in versione **Ms - Dos**, evidenziando, in particolare, le differenze rispetto alla versione Amiga, per consentire ai possessori di quest'ultimo di sperimentare efficacemente le proprie esperienze grafiche anche sulla versione Ms - Dos; ai "dossisti" di conoscere alcune caratteristiche di un pacchetto di indubbio interesse per il loro sistema operativo; a tutti, infine, di conoscere le differenze non solo tra le versioni del pacchetto, ma anche tra le prestazioni dei computer ospiti.

Differenze strutturali

La principale area di differenza tra Deluxe Paint per Amiga e per Ms - Dos è nelle **risoluzioni grafiche**. Come molti sapranno, Ms - Dos prevede i modelli di grafica in base a schede hardware fornite separatamente dal computer: **CGA**, **EGA**, **MCVGA**, **Hercules**, **E-VGA** sono sigle ben note, che indicano non solo differenti standard di risoluzione grafica e gamme cromatiche, bensì anche una varietà di prestazioni tecniche in funzione del costruttore.

In parole povere, una scheda VGA della Paradise con 1024 K di memoria video interna non ha affatto le stesse risoluzioni grafiche (nè eguale velocità operativa) di una scheda VGA della Tseng con 256 K di memoria video. Questo perchè, originariamente, i computer Ms - Dos erano forniti soltanto con schede testuali e/o grafiche a bassa risoluzione o limitatissimo numero di colori (Hercules e CGA, difatti ben standardizzate), mentre EGA, VGA ed altri formati sono stati proposti liberamente dopo l'introduzione di nuove macchine (modelli IBM AT) e non come standard commerciali, per cui molte case produttrici si sono ritrovate ad offrire liberamente risoluzioni e tavolozze cromatiche differenti.

I programmi come DeluxePaint sono dotati di parecchi file aggiuntivi per garantire la compatibilità (teoricamente) con tutte le schede grafiche presenti sul mercato. In ogni caso, càpita che alcune schede non possano dare "il meglio di se stesse" con questo o quel programma, per incompatibilità del software di gestione o semplicemente perchè non sono

Risoluzioni e tavolozze di D. Paint II e III per Amiga:

320 x 512. Interallacciato, 32 colori selezionabili tra 4096, oppure 64 in modo extra halfbrite.

Sfarfallante ("flicker") su monitor normali (1084 o TV), normale se si usa un monitor **multiscan** con scheda **VDE** esterna su Amiga 2000, oppure su Amiga 3000 (dotato della VDE su scheda madre).

640 x 256. 16 colori selezionabili tra 4096.

320 x 256. 32 colori selezionabili tra 4096, 64 in modo "extra halfbrite" (32 colori fondamentali e 32 replica di questi con intensità dimezzate).

640 x 512. Interallacciato, 16 colori selezionabili tra 4096. Sfarfallante ("flicker") su monitor normali (1084 o TV), regolare se si usa un monitor multiscan con scheda VDE esterna su Amiga 2000, oppure su Amiga 3000 (dotato di VDE su scheda madre).

previste dal pacchetto, che non avendo uno standard fisso di riferimento, può comunque incoporare un numero finito di compatibilità (esattamente come avviene per le stampanti grafiche).

Amiga non ha schede grafiche aggiuntive, essendo dotato di Hardware interno per lo scopo. Le risoluzioni disponibili in Deluxe Paint sono perciò fisse e standardizzate, come visibile nella tabella riportata. Il nuovo **ECS** montato di serie solo sugli **Amiga 3000** per il momento, consente altre risoluzioni grafiche, per altro non ancora sfruttate da Deluxe Paint III.

In tutti i casi, la risoluzione grafica ed il numero di colori disponibili dipende anche dalla disponibilità di **memoria** del proprio computer.

Per Ms - Dos è bene avere **640 K** di memoria (almeno) per lavorare tranquillamente con Deluxe Paint in alta risoluzione, mentre per Amiga è bene disporre di almeno **1 Megabyte** di memoria.

Deluxe Paint III per Amiga comprende una serie di funzioni di semplice animazione, delle quali abbiamo già parlato parzialmente nel numero 79 di CCC, basate sulla successione rapida di singoli fotogrammi, sulle "brush" o sul ciclaggio di colori nella tavolozza.

Nella versione Ms - Dos di Deluxe Paint IIe, invece, esiste principalmente il ciclaggio dei colori, ma nessun menu di funzioni di animazione basate sulle brush. Per chi desiderasse sfruttare le animazioni, esiste un programma "gemello" chiamato **Deluxe Animator** che è un pacchetto, ispirato ai criteri di Deluxe Paint, completamente dedicato al compito per il mondo Ms - Dos. Il pacchetto è ovviamente molto più potente, per quanto riguarda le animazioni, di quanto consentano i menu di Deluxe Paint III di Amiga, che a sua volta è invece molto più dotato nel settore "animazioni" del Deluxe Paint IIe per Ms - Dos.

Altre differenze, quasi sempre marginali, si trovano nei menu a discesa, selezionabili premendo il pulsante destro del mouse su Amiga ed il pulsante sinistro (col puntatore collocato sulla barra del menu) in Ms - Dos. Il menu ad icone sulla destra è funzionalmente quasi identico tra le due versioni, fatte salve le differenti implementazioni delle funzioni offerte e delle sagome utilizzate per indicarle.

I menu Ms - Dos

Inizialmente bisogna scegliere quale tipo di risoluzione grafica si desidera. La scelta è determinata dalla scheda grafica montata nel computer. In generale, avendo una VGA, si possono usare anche (quasi) tutte le risoluzioni CGA ed EGA, ma dipende dalla quantità di RAM video montata sulla scheda (256 K, 512 K oppure 1024 K), dalla sua compatibilità effettiva con il programma Deluxe Paint e dalla quantità di memoria centrale libera nell'elaboratore.

Bisogna anche avere installato un **mouse**, collegandolo al PC con una scheda o via seriale, che viene automaticamente riconosciuto (se correttamente installato) da Deluxe Paint all'avviamento; **in mancanza del mouse, il programma non parte** e riporta un messaggio che indica la necessità di un mouse. Il problema non sussiste, ovviamente, con Amiga, che ha il mouse di serie.

I Menu a sipario del programma DOS sono 7: Picture, Brush, Techniques, Stencil, Font, Misc. e Preferences. Il primo comprende tutte le opzioni di gestione generica del lavoro: lettura (**Load**) e salvataggio dei file grafici (**Save**), scambio della pagina corrente con quanto contenuto in un buffer (**Page / Spare**), cancellazione di pagine visualizzate o nel buffer (**Delete**) ed altro ancora.

Di diverso dalla versione Amiga è qui la modalità di selezione della stampa su carta della pagina grafica: con **Choose Pr** è possibile scegliere tra una quantià di stampanti grafiche note al programma, insieme al formato del foglio ed all'altez-

Chi è Dan Silva

Dan Silva, autore di Deluxe Paint per Amiga e Ms - Dos, è un ingegnere meccanico californiano. Imparò a programmare dopo la laurea, trovandosi impegnato in un progetto inerente la progettazione di giunzioni cinematiche di arti robotici.

In seguito lavorò alla NASA come programmatore per la realizzazione di software scientifico: dall'elaborazione di dati provenienti da satelliti artificiali a quelli provenienti da gallerie del vento per lo studio dell'aerodinamica di missili.

Dopo quattro anni alla NASA, si staccò per esplorare il mondo della programmazione a basso livello, lavorando alla **Genrad** su di un sistema a microprocessore che analizzava in tempo reale segnali provenienti da strutture vibranti. Dopo nove mesi, appreso tutto sulla programmazione a basso livello, se ne andò per passare alla **Xerox** System Development, iniziando a lavorare su stazioni grafiche **Star**, dove venne contagiato definitivamente dall'amore per la grafica. Dalla Xerox passò alla **LucasFilm**, dove in un anno e mezzo realizzò un videoeditor chiamato **EditDroid**, ma a tempo perso lavorava in applicazioni grafiche, scrivendo un programma di disegno chiamato **Pixar**, usato da alcuni ingegneri per disegnare circuiti, in pratica il prototipo di Dpaint. In seguito a questo, ebbe una commissione esterna per realizzare un pacchetto di grafica, che chiamò **Doodle** e che incorporava molto dell'attuale Deluxe Paint.

Infine, per la **Electronic Arts** (casa produttrice di Deluxe Paint) lavorò alla creazione degli strumenti di lavoro necessari agli svilppatori per creare software. Il risultato del tool grafico di base che i programmatori della ECA usavano per realizzare le pagine grafiche dei pacchetti (indipendentemente dalla macchina finale sui quali avrebbero poi dovuto funzionare, C/64, Apple, Ms - Dos o Amiga) è, con qualche piccola modifica ovviamente, l'attuale Deluxe Paint che tutti conosciamo.

Il programma, scritto in larga parte in **C** ed in **Assembler** in alcune sezioni critiche, ha subito profonde modifiche: per Amiga sono uscite la versione **I**, **II** e **III** (in varie release ciascuna), mentre per Ms - Dos a tutt'oggi sono disponibili le versioni **II** e **II e**nhanced.

za e larghezza in centimetri (o pollici) della stampa. Segliendo successivamente l'opzione **Print** dallo stesso menu il programma eseguirà le necessarie conversioni e regolazioni per ottenere una perfetta stampa sulla periferica selezionata e configurata con **Choose Pr**.

I file IFF sono scambiabili liberamente tra i mondi Amiga e Ms - Dos, dopo avere, ovviamente, convertito il formato dei dati dei dischi di trasferimento e rispettato le limitazioni basate sul numero di colori e risoluzioni grafiche. In generale bisogna tenere conto delle differenti tavolozze cromatiche (**4096** colori in Amiga, **256** tipicamente in Ms - Dos) e delle differenti risoluzioni video prima di eseguire dei "porting" tra i due mondi di file grafici.

Brush

Il secondo menu, chiamato Brush in ambedue le versioni del pacchetto, sottindende le funzioni di gestione delle brush, ovvero dei **ritagli di schermo** eseguiti dopo avere cliccato sull'icona apposita sulla destra dello schermo (la prima delle cinque file nella versione Ms - Dos) ed avere indicato l'area col mouse. Una volta definita la brush, è possibile da menu salvarla o caricarla da disco (Save e Load), dimezzarne (**Halve**) o raddoppiarne (**Double**) le dimensioni oppure variarle direttamente col mouse (**Stretch**), ruotarla specularmente (**Flip**) lungo la longitudine o latitudine, ruotarla in senso orario (**Rotate**) di vari angoli (90, 180) o liberamente, nonchè eseguire su di essa sfumature cromatiche (**Bend** e **Smear**) in base ai valori di sfumatura definiti nella **Palette**, richiamabile con l'apposita opzione di menu oppure con la pressione del tasto P.

La versione Amiga prevede altre opzioni, molto coreografiche, ma le funzioni di base sono presenti e pressochè identiche in ambedue le versioni. Amiga consente di **trascinare** in tempo reale qualunque brush, mantenendola visualizzata mentre si muove il mouse grazie all'enorme velocità operativa del suo hardware evoluto (in particolare, il Blitter), mentre Ms - Dos risulta particolarmente lento nella gestione delle Brush, specialmente se non si usa un sistema con un processore veloce, non essendo dotato di alcun coprocessore grafico.

Tecniche

Il menu **Techniques** equivale al menu **Mode** della versione Amiga. Prevede alcune opzioni che controllano il modo in cui i colori vengono resi sullo schermo durante le tracciature grafiche. Ad esem-

Che cosa è IFF

Deluxe Paint registra i file grafici nel cosiddetto formato **IFF**. Si tratta di un formato standard nel mondo **Amiga**, ma relativamente poco diffuso nel mondo **Ms - Dos**.

Ciò significa che tutti i pacchetti utilizzanti file grafici nel mondo Amiga (impaginatori, CAD, digitalizzatori...) utilizzano questo formato e possono tranquillamente scambiare i propri dati.

Nel mondo Ms - Dos esistono numerosi formati di memorizzazione, tra loro incompatibili, che costringono spesso ad usare sofisticati programmi di conversione (a volte imponendo decadimenti di qualità) per consentire di trasportare i dati tra un pacchetto e l'altro.

IFF è l'acronimo di "**I**nterchange **F**ile **F**ormat" ed è stato inventato dalla **Electronic Arts** per consentire lo scambio di dati di qualunque tipo (grafici, testi, musica, digitalizzazioni sonore, animazioni, eccetera) tra computer di vario tipo.

Difatti viene utilizzato correntemente non solo in ambiente Amiga (dove è lo standard unico per i file grafici), ma anche sotto Ms - Dos, Macintosh e Unix.

pio, **Smear** e **Shade** sono due effetti che "spargono" il colore in modo da dare sfumatura od effetto di ombreggiatura, rispettivamente. L'uso delle opzioni deve essere sperimentato direttamente, in quanto a parole è difficile spiegare ciò che risulta intuitivo dopo solo pochi minuti di utilizzo pratico. Gli effetti possibili sono molto interessanti, in particolare nella versione Amiga, che, prevedendo costantemente una tavolozza di 4096 colori, consente di effettuare sfumature più delicate ed effetti di ombreggiatura più convincenti della versione Ms - Dos.

Stencil

Lo Stencil è, come il brush, un concetto innovativo nei programmi di grafica creativa. Consente di "chiudere", alle modifiche, uno o più colori, scelti da un apposito requester che appare scegliendo **Make Stencil** dall'apposito menu. Quasi le stesse opzioni sono raggruppate in un menu apposito nella versione Ms - Dos ed in un sottomenu di **Effect** della versione Amiga.

Per usare Stencil si deve scegliere **Stencil On**: appare una tavolozza dove i quadratini colorati indicano i colori "bloccabili" alla pressione del mouse. Un colore bloccato non può essere modificato da alcuna operazione di tracciatura, riempimento o brush sinchè non viene esplicitamente sbloccato cliccando nuovamente con il mouse nel requester Make Stencil.

Altre opzioni del menu consentono di chiudere alle modifiche il colore di primo piano (**Lock Foreground**), di invertire la selezione dei bloccaggi, di caricare o registrare sul disco certe stencil (Load e Save) e di bloccare o liberare lo sfondo (**Background Fix** o Background **Free**). Si noti che il funzionamento di alcune opzioni dipende dalla presenza nel sistema di memoria espansa (a volte anche 100 Kb), quindi non sono operative su tutti i modelli Ms - Dos, ma solo su quelli dotati di particolari configurazioni hardware (processore 286 o 386 e memoria espansa).

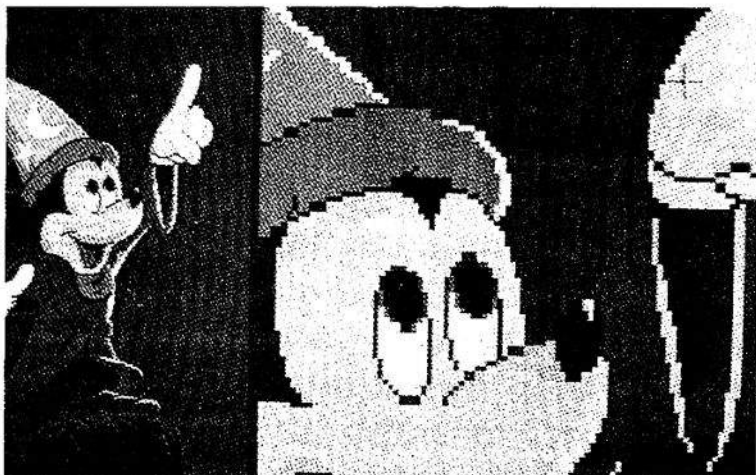
Fonti

Deluxe Paint consente di usare una quantità di fonti di caratteri, di cui è fornita una buona serie già sui dischi a corredo, in particolare nella versione Ms - Dos.

Numerose softhouse commercializzano dischi aggiuntivi di fonti da utilizzare espressamente con Deluxe Paint.

Tutti i colori di Ms - Dos

Ecco la tabella delle risoluzioni grafiche e del numero di colori utilizzabili con Deluxe Paint II Enhanced per Ms-Dos in funzione della scheda grafica inserita nel computer:

Scheda	Risoluz.	Colori
CGA	320 x 200	4
CGA	640 x 200	2
EGA	320 x 200	16
EGA	640 x 200	16
EGA	640 x 350	16
MCGA	320 x 200	256
MCGA	640 x 480	2
VGA	640 x 200	16
VGA	640 x 350	16
VGA	640 x 480	16
Hercules	720 x 348	2
Tandy	320 x 200	16
Amstrad	640 x 200	16
E-VGA	640 x 400	256
E-VGA	640 x 480	256
E-VGA	800 x 600	2
E-VGA	800 x 600	16
E-VGA	800 x 600	256
E-VGA	1024 x 768	2
E-VGA	1024 x 768	16

Le fonti vanno caricate da disco tramite un apposito requester che appare scegliendo **Choose Font** dall'apposito menu, oppure cliccando con il pulsante destro del mouse sulla seconda icona (riportante una "**A**") della sesta riga del menu verticale sulla destra dello schermo di lavoro. Una volta scelta una fonte di caratteri, e caricata da disco, bisogna indicarne la dimensione (espressa in pixel verticali rispetto ad una teorica linea di base, definita da chi ha disegnato il set di caratteri) tra quelle possibili, indicate nel menu Font. Nella versione Ms - Dos le fonti variano tra gli **8** ed i **96** pixel di altezza, in **9** suddivisioni, mentre nella versione Amiga le fonti sono "continue" e non procedono per dimensioni fisse nè hanno un limite fisico di altezza, se non in base alle limitazioni dell'hardware.

Disponendo di una scheda grafica adatta, con 256 colori, nella versione Ms - Dos è consentito caricare ed usare fonti multicolore, ovvero dotate di più di due colori nell'ossatura grafica. La versione Amiga prevede l'uso delle fonti a colori se si è lanciato, prima di Deluxe Paint, un apposito programma (**Color Font**) da Workbench. I file di fonti a colori sono differenti da quelli normali. Inoltre, il formato delle fonti a colori **non** è trasferibile tra mondo Amiga e Ms - Dos..

Miscellanea

Il menu **Misc.** della versione Ms - Dos raggruppa varie opzioni presenti come sottomenu nella versione Amiga, ma a parità di nome di funzione, le caratteristiche sono identiche.

Info Bar visualizza inferiormente una barra con varie informazioni di lavoro, aggiornate in tempo reale. Scegliendo anche **Coordinates** dallo stesso menu, si ottiene la visualizzazione, nella barra inferiore, delle coordinate corrispondenti alla posizione del puntatore del mouse, in pixel od in centimetri secondo la scala regolata.

La funzione **Anti-Alias** usa un particolare algoritmo di distribuzione dei colori che minimizza l'effetto di *aliasing*, mentre **Colorize** usa il colore di tracciatura come "colorizzante" in base alla **Tecnica** (da omonimo menu) selezionata. **Translucent** usa il colore di background per scoprire il colore sottostante durante la tracciatura.

La funzione **Cycle Colors** effettua la rotazione ciclica dei colori indicati con la velocità stabilita, secondo i settaggi definiti nell'apposito requester di tavolozza (visualizzabile premendo il tasto **P** o selezionando **Misc / Palette**). E' questa che consente semplici funzioni di animazione, insieme ad un rudimentale sistema di commutazione di pagine grafiche.

Preferenze

Il menu Preferences raggruppa, in ambedue le versioni, delle funzioni che consentono essenzialmente di configurare il programma a che lavori in un modo a noi congeniale. Nella versione Ms - Dos le funzioni qui raggruppate sono più numerose, ma in generale sono tutte presenti anche nella versione Amiga, sparse per altri menu.

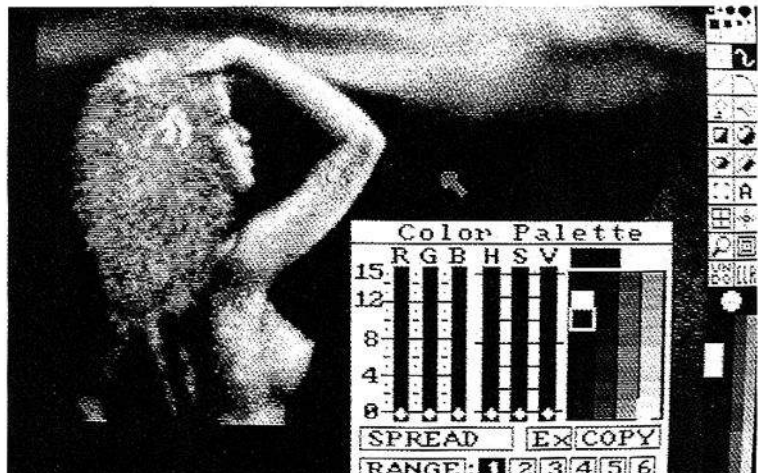
Ad esempio, l'opzione **Fast Feedback** inibisce la tracciatura dell'eventuale brush ad ogni spostamento del mouse, accelerando il lavoro ed evitando disturbo, in particolare durante le operazioni di tracciatura speculare (secondo gadget della nona fila sulla destra dello schermo). Da qui è anche possibile scegliere le unità di misura (pollici, centimetri o punti), il drive da usare per conservare dati di lavoro (**Spare Page Prefs**) ed altro ancora.

Il menu a icone

Il menu ad icone sulla destra prevede funzioni aggiuntive, praticamente identiche tra le due versioni. La versione Amiga consente di selezionare direttamente da qui la dimensione del pixel di tracciatura, mentre la versione Ms - Dos richiede di premere il pulsante destro del mouse sulla prima icona della prima fila: portando sullo schermo il pixel, e muovendo il mouse mantenendo premuto il pulsante sinistro, è possibile variare la dimensione del pixel di tracciatura; rilasciando il pulsante la sagoma rotonda definita sarà usata per le successive tracciature.

In generale, tutte le opzioni del menu prevedono requester di regolazione del loro modo di funzionamento, attivabile cliccando sol pulsante destro del mouse sulla icona stessa.

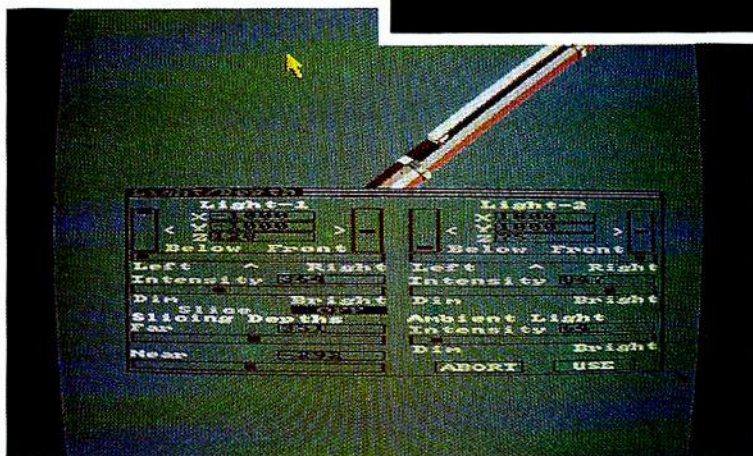
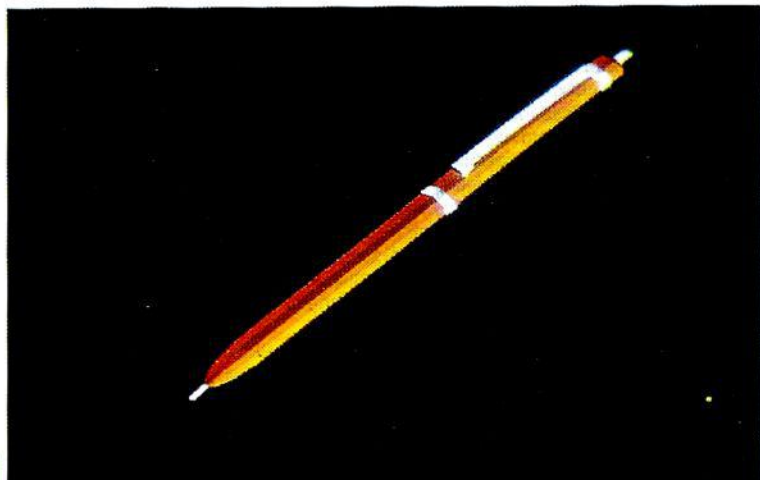
Ad esempio, cliccando la prima icona della seconda fila si possono tirare linee rette col mouse: cliccando su di essa col pulsante destro appare un requester che consente di regolare l'eventuale punteggiatura delle linee tracciate.

La funzione di ingrandimento (icona a lente) funziona in modo diverso nelle due versioni: in quella di Amiga il fattore di ingrandimento può essere variato continuamente, cliccando sul gadget accanto alla lente di ingrandimento sul menu di destra; nella versione Ms - Dos l'ingrandimento è fisso. In ambedue, comunque, si possono usare i tasti cursore per portarsi sulla superficie ingrandita.

di Gregor Samsa

TRE DIMENSIONI TUTTE DA COSTRUIRE

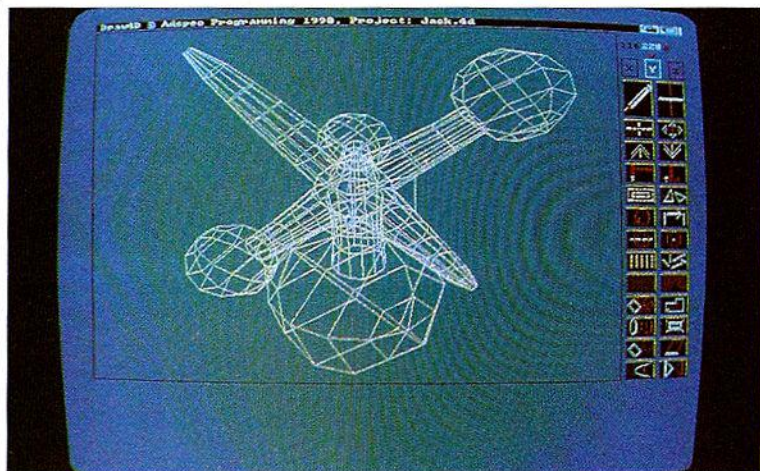
Un super programma per disegnare professionalmente con Amiga

La grafica professionale, o anche solo semiprofessionale (differenza, in verità, sottile) ha trovato in **Amiga**, sin dalla sua comparsa, un notevole strumento di ausilio. Tanto nel settore della tradizionale illustrazione bidimensionale, che in quello più complesso ma di maggiore impatto visivo della cosiddetta fascia **3D**. Sigla che, come tutti sapranno, sta ad indicare l'aggiunta della terza dimensione, quella spaziale.

Creare l'illusione ottica di un oggetto tridimensionale può in apparenza sembrare semplice, ed in teoria può anche esserlo per un provetto artista: l'uso della prospettiva, associato a un oculato gioco di ombre, può far miracoli anche adoperando normali programmi grafici come **Deluxe Paint**. Tuttavia, occorre prendere in considerazione due argomenti. Il primo, più ovvio, che non tutti sono così bravi a realizzare con mezzi propri un disegno in 3D. Ed inoltre: così come un qualunque tool grafico ci viene incontro automatizzando quanto più possibile la creazione di elementi grafici, quali poligoni, cerchi, colorazioni, ombreggiature, eccetera, perché non affidarsi allo stesso Amiga per farsi dare una mano nell'arduo **passaggio da una immagine bidimensionale ad una struttura "solida"**?

Se, poi, al concetto di immagine si aggiunge quello di animazione (sempre

in 3D), ecco che l'uso di programmi *ad hoc* diventa pressoché obbligato, a meno di non voler trascorrere il resto della propria vita nella realizzazione di una marea di "frame" con diversa angolazione di luce, diversa prospettiva, eccetera eccetera.

Tra il software dedicato a queste mansioni, va annoverato **Draw4d**, della statunitense **Adspec**, che si caratterizza soprattutto per un approccio alla materia decisamente più semplice rispetto ai suoi diretti rivali, pur non raggiungendone in alcuni aspetti la stessa levatura. In generale, si può infatti inquadrare Draw4d come un programma in grado di **generare rapidamente figure solide tridimensionali partendo da una immagine piana, con la facoltà di produrre e mostrare un' animazione delle stesse**. Quest'ultimo aspetto è, però, funzionale soprattutto alla elaborazione dei "frame" (le singole schermate che compongono un'animazione), mentre la visualizzazione in senso stretto di un oggetto in 3D animato è ridotta all'essenziale, giusto per controllare l'andamento del lavoro.

Draw 4d, insomma, può considerarsi uno strumento completo per ciò che riguarda la creazione di figure solide, e complementare ad altro software (per esempio **LightWave 3d**, **Animation**, ecc.) nel settore dell'animazione. Di particolare importanza, inoltre, la dichiarata attenzione rivolta al mondo **Dtp** (**D**esk **T**op **P**ublishing), ove le sue realizzazioni possono trovare impiego ideale. Non a caso tutte le immagini generate da Draw4d possono essere memorizzate tanto in comune formato **IFF**, quanto come CLIPS di Professional Draw, in grado quindi di essere trattate come grafica vettoriale. Una schermata prodotta da programmi dedicati alla grafica bitmap (come **DPaint**, per intenderci), per quanto perfetta, non può certo essere manipolata secondo libere esigenze di impaginazione (allargata, ristretta, ecc.) senza perdere di qualità, e diventa quindi indispensabile l'uso di software che generi una grafica adatta allo scopo.

In azione

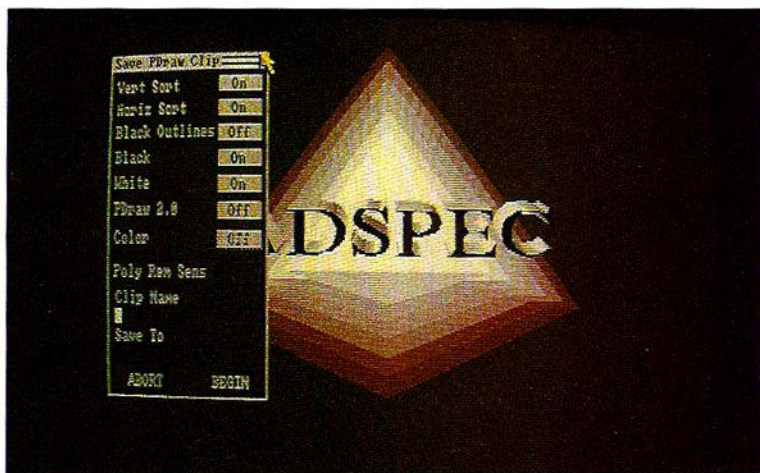
Vediamo più da vicino come opera Draw4d, simulando tutti i passaggi per giungere dalla schermata iniziale fino alla realizzazione di un oggetto solido.

Gli ambienti di lavoro sono in pratica due. Uno, quello principale, è caratterizzato subito dopo il lancio del programma dalla presenza al centro dello schermo di tre assi (X, Y e Z) due dei quali ruotano attorno ad una linea verticale.

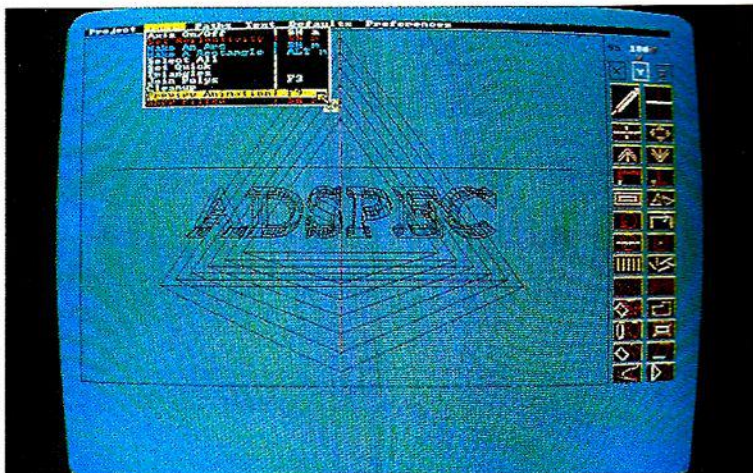
Premendo la barra spaziatrice si arresta il movimento, e si può procedere ad *appoggiare* sull'asse orizzontale un primo *oggetto*. Tutta la grafica di questa prima fase di edificazione è basata sull'uso di poligoni, ma soprattutto dei punti che ne identificano i vertici. In tal senso anche una sfera è vista come un poligono, composto da tanti più lati quanto maggiore si desidera la definizione della curva.

Una realizzazione a mano libera risulta piuttosto complessa, in quanto sarebbe necessario agire sui soli punti di ogni "lato" per dar forma all'oggetto.

Ma, naturalmente, viene in aiuto il programma, grazie ad alcune opzioni del **menu Tools** adatte allo scopo. Basta scegliere, per esempio, **Make a Rectangle**, ed apparirà un requester per definirne le dimensioni. Dopo di ciò, un bel rettangolo verrà centrato tra gli assi (a meno che non si sia scelto un posiziona-

mento diverso). La figura sarà ancora bidimensionale, ma si può già provare a vederla sotto diverse angolazioni.

Agendo sul tastierino numerico, si può infatti attivare la rotazione anche contemporanea di tutti e tre gli assi. I tasti verticali di centro (**2**, **5**, **8**) fermano il movimento, mentre quelli laterali lo attivano in un senso o nell'altro.

Più volte si "pesta" sul tasto, più veloce diventa il movimento, mentre la barra di spazio fa tornare alle condizioni iniziali. Se si desiderano spostamenti veloci, soprattutto quando immagini molto complesse rallentano alquanto i movimenti, si può sempre scegliere **Quick Move** dal menu **Preferences**, che ridurrà all'indispensabile la visualizzazione del poligono, accelerandone gli spostamenti rotatori. Una volta raggiunta la posizione voluta, si può poi tornare alle condizioni originarie riselezionando la stessa opzione.

Nelle prime fasi la rotazione ha ancora poco senso, ma diventa di notevole effetto dopo aver dato "spessore" al poligono. Per farlo, la manovra più semplice consiste nell'attivare il poligono clickando nel suo **punto di origine** (così facendo questo assumerà una colorazione bianca), quindi selezionare l'apposita **icona** presente sul bordo destro dello schermo. Per precisarne i parametri, si potrà clickare la stessa col pulsante destro, oppure selezionare dal menu **Defaults** la voce **Extrude Defaults**, o ancora premendo

Ctrl + E. Agendo nuovamente sul tastierino numerico per far ruotare l'ex rettangolo, si potrà vedere come questo sia, in effetti, diventato un parallelepipedo.

La rotazione servirà, quindi, non solo per visionare ed eventualmente modificare il poligono, ma anche per posizionare la figura in funzione del cosiddetto "rendering", ovvero l'ultima fase. Prima di arrivarci, è però il caso di accennare ad almeno qualcuna delle numerosissime potenzialità che il programma ancora offre.

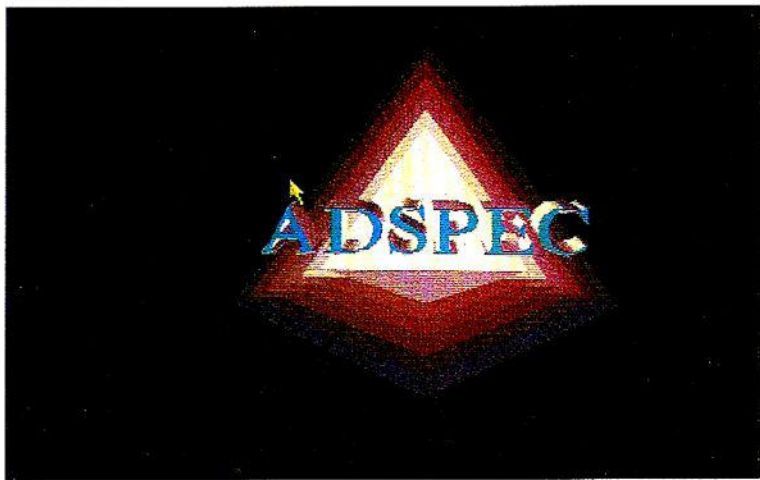
Oltre alla creazione automatica dei poligoni ed al loro "extrude" (= resa dello spessore), è infatti possibile ottenere proiezioni, duplicazioni, associazioni di più poligoni fino a formare figure anche molto complesse, e una miriade di altre chance offerte dai nutriti menu di Draw4d.

Particolarmente interessante la capacità di adoperare per l'Extrude una figura predefinita, ovvero un altro poligono, che diventerà la sezione dello spessore. Inoltre, e questa è davvero succosa, piuttosto che poligoni è possibile adoperare delle lettere dell'alfabeto, singolarmente o connesse in stringhe, che potranno poi essere salvate come **Font di caratteri tridimensionali**!

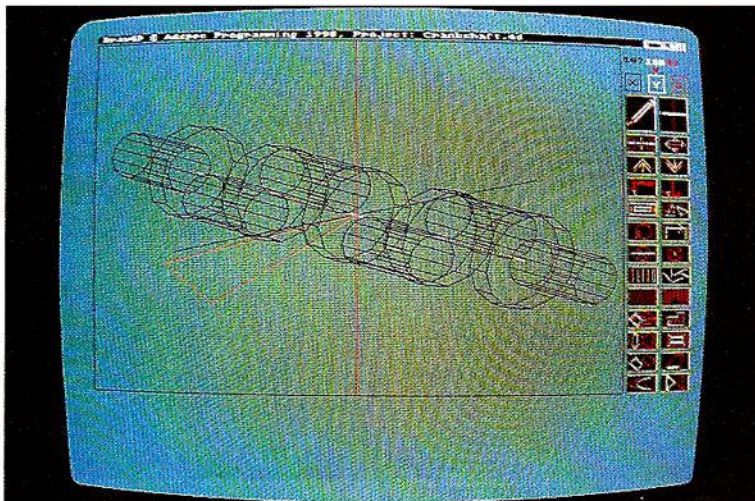
Due *font* sono già incluse nel disco principale del programma, e possono facilmente essere caricate ed adoperate per propri esperimenti in 3D. Un carattere, infatti, può essere trattato esattamente come qualunque altro poligono, fino a raggiungere effetti di tutto rispetto.

E fu 3D

Fin qui, l'oggetto (o carattere che sia) viene trattato e visualizzato nelle sue linee essenziali, o meglio "punti" essenziali, cosa che ne rende agevole la modifica non solo prospettica, ma anche nelle dimensioni generali. Tramite la funzione di **Zoom**, è infatti possibile "avvicinare" o "allontanare" il soggetto senza con que-

sto perderne alcun dettaglio. Comunque, una volta raggiunto quello che sarà (salvo ulteriori modifiche) il posizionamento finale dell'imagine, si può passare alla fase di **Rendering**, ovvero la creazione effettiva dell'equivalente schermata bitmap, non più limitata alle sue linee essenziali.

Tutto ciò che occorre fare, è selezionare **Show Filled** dal menu **Tools** (o premere Shift + F9), e aspettare qualche secondo. Si entrerà in nuovo ambiente di lavoro, nel quale verrà visualizzata l'immagine pressoché definitiva, sulla quale sarà comunque ancora possibile qualche intervento: scelta della risoluzione (anche **Ham**!), della palette colore, e resa "piena" o meno del solido.

Come ogni rendering che si rispetti, è poi possibile impostare la simulazione di **due fonti luminose**, che a seconda dell'angolazione scelta creeranno automaticamente un gioco di luci ed ombre essenziale per l'illusione di tridimensionalità.

Se l'effetto finale non soddisfa, si può sempre invertire l'effetto delle luci, modificare il cosiddetto **Dithering**, o tornare all'ambiente principale di lavoro e "rifare il trucco" all'immagine.

Volendo, in questa fase si può anche aggiungere uno sfondo caricando dall'esterno una qualunque schermata grafica IFF, ma l'operazione più importante, per quanto possa sembrare banale, resta il salvataggio su periferica.

L'intero risultato dei nostri sforzi in 3D può essere trattato come una qualunque schermata grafica, e come tale salvata in standard IFF per essere poi riutilizzata tramite programmi che gestiscono questo formato.

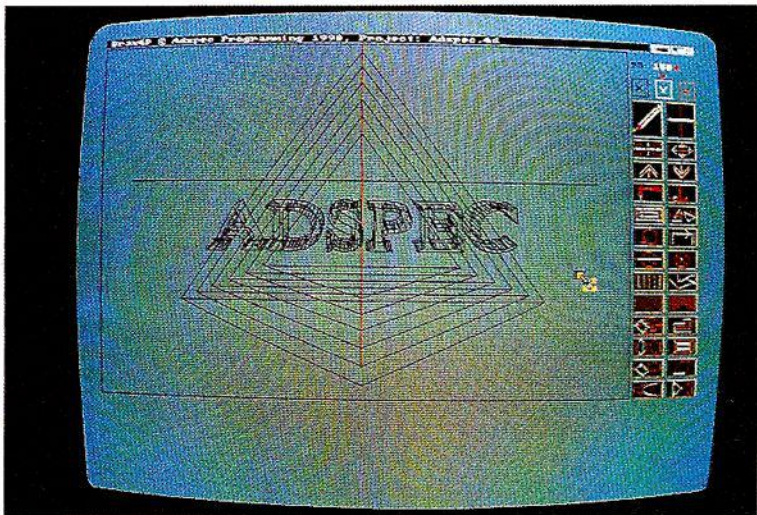
Se invece si è impostato il lavoro in funzione di **Frame** per un'animazione, è disponibile anche un **Save** in formato **Anim**, anche questo uno standard di larga diffusione.

In questa eventualità, i tempi di attesa si fanno però decisamente lunghi, a meno di non disporre di schede acceleratrici.

Altro formato selezionabile, il **Clip** di Professional Draw. In questo caso, è poi possibile rimaneggiare ulteriormente gli oggetti adattandoli alle esigenze... editoriali consentite dal programma della Gold Disk.

Se si esclude il suddetto Professional Draw, non è implementata alcuna altra compatibilità, neanche con altri programmi dedicati al rendering, cosa che fa sentire la sua mancanza.

Sarebbe ottimale, infatti, un trattamento iniziale con Draw4d, considerata la sua semplicità e rapidità di esecuzione, magari da continuare in altri ambienti (per esempio) per incrementare il gioco di luci, come già detto limitato a soli due punti fissi di riferimento, e senza la possibilità di aggiungere terzi elementi che lancino la loro ombra sugli oggetti preesistenti.

Tutto questo volendo proprio cercare qualche limite in un programma che, in definitiva, si propone come il più adatto per chi si avvicina al mondo del software 3d, ma anche uno dei più "sbrigativi" per chi già ha esperienza nel settore.

di Domenico Pavone

DAL SUONO ALL'IMMAGINE

Due accessori di tutto rispetto per ampliare le già notevoli potenzialità grafiche di Amiga

Non vi è alcun dubbio: è il loro momento. Se si escludono i comuni accessori dedicati alla memoria (interna o di massa), che possono essere considerati normale routine nell'evoluzione di un computer, in questo periodo stanno vivendo un momento magico i **digitalizzatori** di ogni tipo riservati ad Amiga.

In queste stesse pagine ci siamo occupati, non molto tempo fa, di un **Audio Digitizer**. A soli pochi mesi di distanza ne rivediamo un altro, della stessa fascia di prezzo, ma dalle prestazioni qualitativamente superiori. E visto che non è solo la musica (o il suono in senso lato) ad essere interessato all'evoluzione di cui sopra, diamo anche un'occhiata ad un oggettino davvero simpatico: uno **scanner** manuale per digitalizzare immagini statiche, anche questo appartenente ad una fascia di prezzo assolutamente accessibile (attorno alle **450 K lire**).

La correlazione con questo tipo di hardware non è così lampante, ma vedremo anche in azione un mouse un po' diverso dal solito, eventualmente da affiancare a quello comunque valido che mamma Commodore fornisce in dotazione al nostro computer preferito.

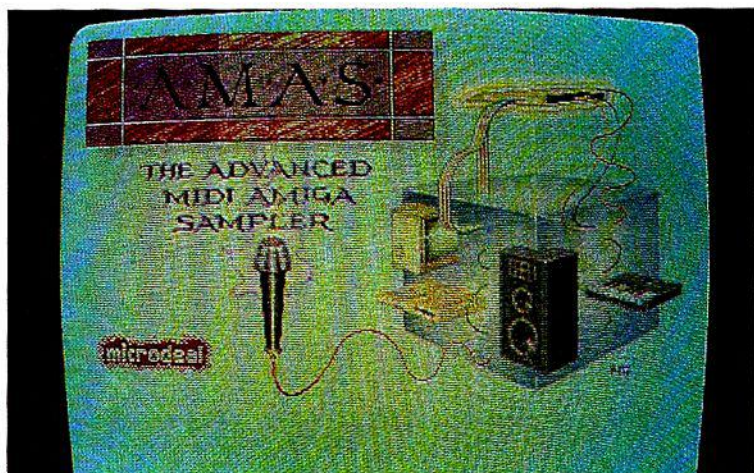
Si sta parlando di Amiga, naturalmente...

A.M.A.S. Sampler

Far riprodurre un qualunque brano musicale ad Amiga, naturalmente in stereo, abbiamo già visto come non sia impresa proibitiva. Si può affrontare il problema sfruttando unicamente delle risorse software, manovrando opportunamente strumenti come **Sonix, MusixX**, o il diffusissimo **Sound Tracker** di pubblico dominio. Tutti programmi di livello molto sofisticato, ma non così facili da spremere a fondo, soprattutto per chi non ha una certa dimestichezza con la notazione musicale. Traduzione: la maggior parte di noi.

Tutto cambia se ci si rivolge a quanto messo a disposizione dalla produzione hardware. Una magica scatoletta da collegare al computer e ad una qualunque fonte sonora, ed il gioco è fatto... o quasi. Si sta parlando, come ovvio, di scatolette non certo dal contenuto commestibile, ma di digitalizzatori audio. A.M.A.S. appartiene, appunto, a questa categoria, come espresso dallo sviluppo del suo acronimo: **A**dvanced **M**idi **A**miga **S**ampler, prodotto dall'inglese **Microdeal**.

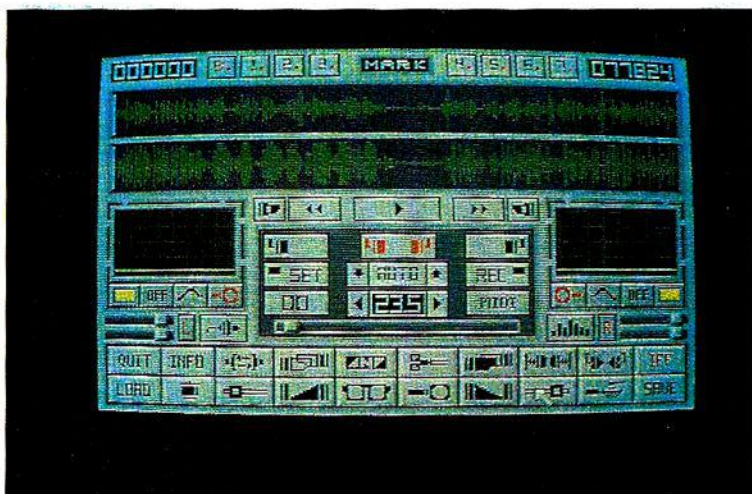
Lo spartano contenitore a cuneo, dal quale fuoriescono due connettori da infilare nelle porte seriale e parallela di Amiga, potrebbe trarre in inganno un osservatore superficiale, ma le sue prestazioni possono raggiungere livelli davvero professionali: non solo in rapporto alla digitalizzazione di brani musicali o campionamento di suono, ma soprattutto per la sua capacità di connettersi in ingresso o in uscita ad una interfaccia **Midi**, per la quale Amas dispone di moltissime opzioni. Sfruttabili, come ovvio, da una utenza musicalmente più specializzata.

Per tutti gli altri, la semplicità di utilizzo del digitalizzatore, connessa ad innumerevoli e intuitive manipolazioni software, può comunque rappresentare un ottimo motivo di soddisfazione, soprattutto nella

In vendita presso
FLOPPERIA
V.le Monte Nero, 15
20135 - MILANO
Tel 02 / 55.18.04.84

personalizzazione di propri floppy o programmi.

Prima di addentrarsi nelle caratteristiche s/w di Amas, va ancora citata la sua semplicità di prelevamento del suono, grazie a due ingressi, una volta tanto, decisamente standard, uno per ogni singolo canale stereo, da collegare (per esempio) all'uscita cuffia di un hi-fi, di un tape, o anche di un walkman. Non manca poi un altro ingresso per un eventuale microfono, anche in questo caso con attacco a jack facilmente reperibile.

Il livello del suono in entrata è molto importante, ma questo può essere facilmente controllato via software, ed adattato di conseguenza. Il package comprende infatti un floppy che, una volta lanciato, fa accedere ad una unica schermata di lavoro estremamente completa e gestibile interamente con il solo mouse. Il funzionamento generale è di una semplicità disarmante, tanto da rendere quasi superfluo il manuale, peraltro abbastanza esauriente. Una volta effettuate le connessioni, il primo passo consiste nella regolazione del volume, ma è anche possibile (cosa non implementata, ad esempio, nel digitizer esaminato alcuni mesi fa) un ascolto per così dire "in diretta" del brano in esecuzione sull'apparecchio collegato.

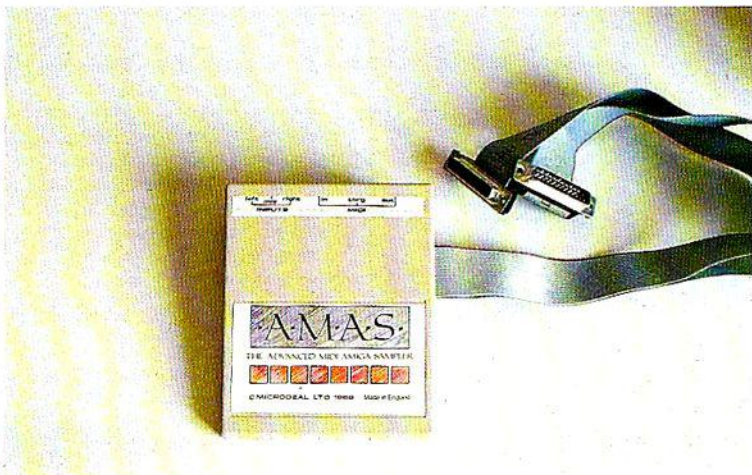
Per il controllo del livello di volume sono presenti due oscilloscopi s/w, uno per canale, che mostrano l'intensità del suono in input, sul quale è possibile agire entro limiti accettabili muovendo col mouse due cursori posti al disotto dei

minischermi degli oscilloscopi. Come ovvio, una prima regolazione va comunque effettuata sul volume generale della fonte, fino ad ottenere una forma d'onda che occupi più o meno l'intera altezza degli oscilloscopi. Nei riquadri riservati ai due canali, sono inoltre presenti due *pulsanti* con mansioni differenti. Agendo su quello collocato accanto alla **"L"** (left, ovvero canale sinistro) lo schermo viene oscurato, e si ha la riproduzione (dall'audio del monitor) di quanto sta emettendo la fonte sonora. Per inciso, con una fedeltà notevole, incrementabile attivando un pulsante elimina-filtro presente nel pannello centrale della schermata.

Simmetricamente disposto nella sezione riservata al canale destro (**"R"**, ovvero right), l'altro pulsante attiva invece un **analizzatore di spettro**, che trasforma la metà superiore del video in un indicatore in tempo reale delle frequenze effettive del brano che si sta ascoltando, e che diventerà il pezzo digitalizzato.

Le principali operazioni di riascolto e registrazione, sono svolte dal pannello centrale. In particolare è possibile selezionare (tasto **Set**) le dimensioni del brano in rapporto alle due bande superiori: si può scegliere di operare solo su quanto contenuto (o che sarà contenuto) nella parte visualizzata, o su interi "banchi", ovvero su quanto è globalmente registrabile (o già registrato).

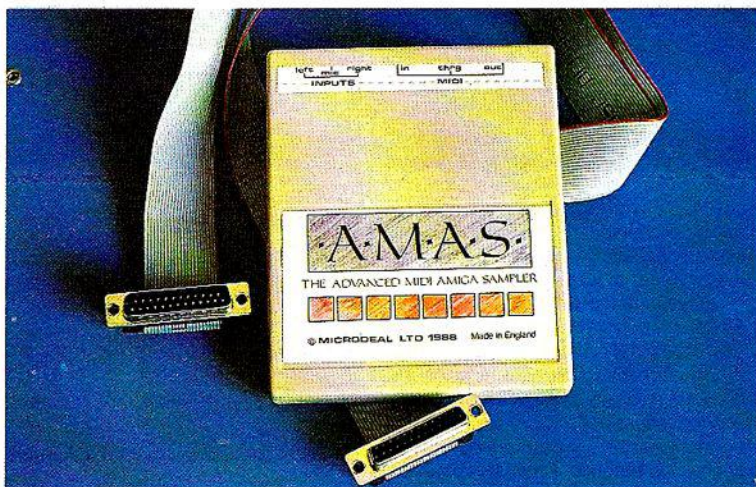
E ancora: si può consentire l'input di solo un canale o di entrambi, e scegliere la "velocità" di registrazione, che influisce nettamente sulla qualità della riproduzione di un brano digitalizzato.

Anche in questo caso, è richiesto solo l'uso a suon di mouse di un cursore o delle freccette poste a lato di un mini display.

Per la **registrazione** vera e propria, non c'è altro da fare che clickare sul tastino **Rec** (difficile, vero?). In più, attivando l'opzione **Auto**, si ha una facilitazione davvero molto gradita: l'operazione inizia solo quando è presente un suono in input, non prima. Si pensi per esempio all'inizio di un brano musicale su disco (o su nastro). Grazie a questo automatismo, si può adagiare la puntina del

giradischi nel solco vuoto tra due brani e attivare Rec col mouse: il brano verrà digitalizzato dal suo effettivo inizio, privo del "silenzio" che lo precede.

Il riascolto è affidato poi al consueto tasto **Play** in emulazione di un normale tape, ed è anche possibile un **ritorno veloce** in avanti o all'indietro, nonché il portarsi all'inizio o alla fine del brano. Attivando poi i tasti L ed R (o anche solo uno dei due), si può avere un ascolto ciclico del brano (o del solo canale selezionato), che ricomincerà ad essere eseguito in un loop infinito.

A queste manovre di base, Amas associa, inoltre, la capacità di apportare una serie di modifiche al pezzo "catturato", che può poi essere salvato su periferica in formato **IFF** oppure **Raw** (file binario), a seconda delle esigenze. Tutta la parte bassa dello schermo è infatti occupata da una serie di icone che implementano gli effetti più svariati: dalla semplice **inversione** dei due canali stereo ad un loro **missaggio**, che può avvenire anche tra diverse porzioni dello stesso canale. E ancora si possono **eliminare** tratti del brano, la cui visualizzazione è sempre attiva in forma di frequenze nelle due bande superiori dello schermo. Gli effetti più interessanti sono forse i cosiddetti **Fadein** e **Fadeout**, che forzano un inizio (o una fine) graduale del suono, o ancora la possibilità di "far passare" un suono da un canale all'altro in progressione.

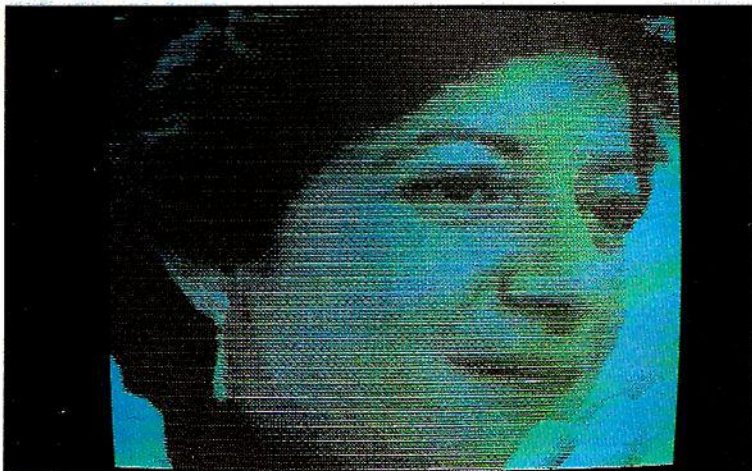
Per intenderci: se si fosse registrato il rombo di una moto o di un motorino opportunamente trattato (il suono, non il motorino...), con l'opzione **Bounce** sarebbe possibile far emettere il suono dallo stereo in modo da simulare il passaggio della moto da una direzione verso l'altra (in realtà da un canale all'altro). Senza montarsi la testa, però: non ci si illuda di diventare più bravi dei Pink Floyd...

E non sarebbe finita qui. Le opzioni sono davvero tante, pur senza entrare nel merito del Midi. Si può ancora accennare alla disponibilità (memoria permettendo) di **8 banchi** sui quali lavorare, copiando e spostando brani dall'uno all'altro, o l'inserimento di un **filtraggio selettivo** (ma definitivo, in questo caso) delle frequenze alte nel brano da salvare.

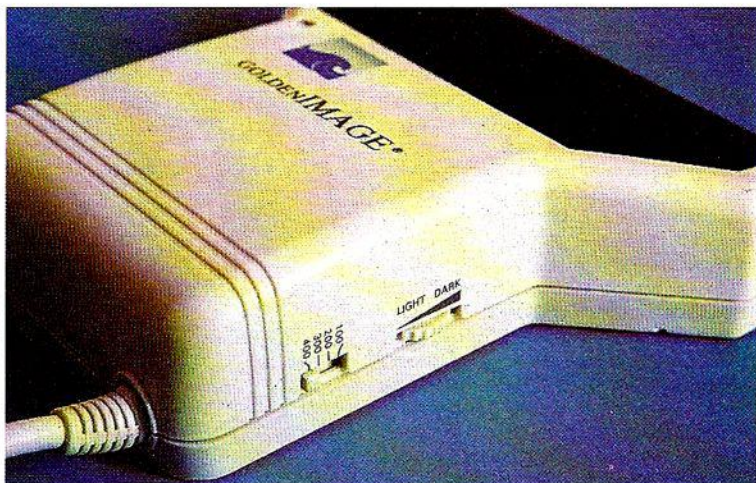
Un A.m.a.s. in definitiva tutto da scoprire, ma soprattutto da provare.

Mouse ottico

Parlare di mouse in casa Amiga non è impresa facile, se si considera l'automatismo nell'acquisizione del connubio computer - mouse. Chi penserebbe di prendere un secondo mouse, quando ne viene già fornito uno in dotazione?

Il problema tutt'al più potrebbe porsi in caso di sostituzione obbligata (leggi: mouse in fumo), ma difficilmente in altre circostanze. A meno che non si prospetti l'acquisto di una periferica con caratteristiche molto particolari, che in realtà modifichino il normale concetto d'uso di un mouse: **Track Ball**, **Mouse - Joystick**, **tavolette** grafiche, e chi più ne ha più me metta.

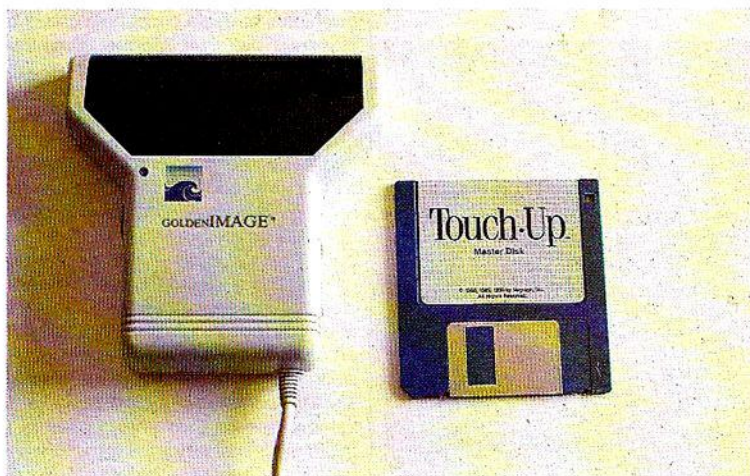
Pur senza giungere a veri stravolgimenti del concetto di mouse, qualcosa di nuovo lo propone tuttavia la **Golden Image**, con un prodotto fruibile dagli utenti di ogni tipo di computer: un **mouse ottico** del quale esistono versioni anche per Atari, Amstrad e **Ms - Dos compatibili**. La definizione dice già tutto sulla novità che lo differenzia dal normale "attrezzo" di origine commodoriana, e basta una prima occhiata superficiale per notarlo: sulla faccia inferiore manca la classica sfera di scorrimento, al posto della quale sembra esserci solo una incavatura di minime dimensioni fornita di **tre fori**, uno centrale e due sulle pareti del piccolo "cratere".

In realtà non si tratta di semplici fori, ma di minuti strumenti in grado di emettere una debole (per l'occhio umano) luminosità, e di percepirne le variazioni del riflesso su una superficie adatta. Variazioni che, come con qualunque altro mouse, si verificano mettendo in movimento lo strumento. Su questa premessa, appare ovvio come l'uso del mouse sia indissolubilmente legato a quello di una **base adatta** allo scopo, del resto inclusa nella confezione: la superficie piana (e di solito monocroma) di un tavolo, non riuscirebbe ad attivare il pointer sullo schermo di Amiga. L'assenza della sfera, come intuibile, dà al mouse una leggerezza notevole, da intendersi non solo in termini di peso ma anche di facilità di scorrimento; al punto che, pur se improponibile come prassi normale, il pointer viene attivato anche mantenendo il mouse appena al di sopra della superficie di lavoro. In ogni caso il minor attrito sulla base giova molto alla sensibilità della manovra, che si manifesta soprattutto in applicazioni grafiche che richiedano una maggiore definizione.

Per quanto possa sembrare di secondaria importanza, non va sottovalutato inoltre l'aspetto pulizia. Non solo minore esigenza di frequenti interventi, ma soprattutto estrema semplicità nel metterli in atto. Senza una sfera che trasferisce all'interno quanto incontrato sulla (spesso... immonda) superficie di lavoro, è sufficiente un banale straccio per eliminare eventuali "scorie", senza la necessità di alcun intervento all'interno del mouse.

Esternamente si presenta con design molto gradevole e slim, anche se la snellezza obbliga la mano ad una posizione leggermente diversa, più obliqua rispetto a quella imposta dal normale mouse di Amiga. In realtà si parla comunque di 11 centimetri di lunghezza per 5 di larghezza, dimensione che la raffinata linea tondeggiante e sfilata fa apparire ancora più esigua.

Il mouse è fornito di **tre tasti** che fanno corpo unico con la superficie di appoggio della mano, disposti sulla estremità anteriore. Il tasto di mezzo, su Amiga, praticamente non serve a nulla, a meno che non si adoperino particolari programmi di ambiente "alieno" fatti girare in emulazione. Eventualità invero piuttosto esigua.

L'interfacciamento con Amiga è totale, nel senso che non necessita di driver particolari: basta inserire il connettore nella solita porta riservata al mouse, e tutto è pronto. La sensibilità nella percezione del movimento è garantita da una risoluzione di **250 Dpi** (Punti per inch) unita ad una velocità di **500 mm per secondo**, e una "prova su strada" con vario software grafico e giochereccio non ha evidenziato lacune di sorta.

La base sulla quale far scorrere il mouse, sebbene più leggera dei comuni "panni" di uso analogo, risulta ben ancorabile al piano di lavoro grazie ad uno strato spugnoso che ne garantisce l'adesione. Le dimensioni di questa (cm. 18 x 23) sono appena sufficienti a consentire l'escursione di tutto lo schermo, soprattutto in senso verticale, ma anche in questo caso senza che la cosa crei reali problemi.

Come già accennato, l'uso dell'Optical Mouse **Golden Image GI-1000** risulta particolarmente indicato nelle applicazioni grafiche a mano libera. Giusto per non restare sul vago, basti pensare al **Deluxe Paint**: adottando la tecnica di mantenere pressato il pulsante sinistro, ma facendo poggiare appena sul panno dedicato solo la parte posteriore del mouse, con l'anteriore sollevata, una banale operazione come apporre sullo schermo la propria firma in corsivo diventa elementare, e soprattutto conforme alla propria reale... calligrafia. Chi ha provato a compiere

una simile operazione con un normale mouse opto-meccanico sa bene quanto sia in realtà complesso ottenere qualcosa del genere (e il mouse di Amiga non è certo da disprezzare), a meno di non usare la penna ottica di una tavoletta grafica.

In conclusione, il mouse della Golden Image desta un'ottima impressione, pienamente avvalorata dai fatti, tanto da non far sembrare così azzardata l'idea di avere un secondo mouse a disposizione, da affiancare al topastro commodoriano topastro.

Purché ottico, s'intende...

Hand Scanner JS-105-1M

Ancora dalla **Golden Image**, viene commercializzato un prodotto di quelli che possono far gola al preponderante settore amatoriale dell'utenza Amiga. Si tratta di un piccolo scanner manuale, dal costo decisamente accessibile.

Per chi non sapesse che cosa sia uno scanner di immagini, diremo brevemente che si tratta di una periferica in grado di trasferire dalla carta allo schermo del monitor qualunque disegno, foto, vignetta, o anche testo. Il tutto automaticamente, secondo un meccanismo che si può paragonare alla fotocopia di un documento.

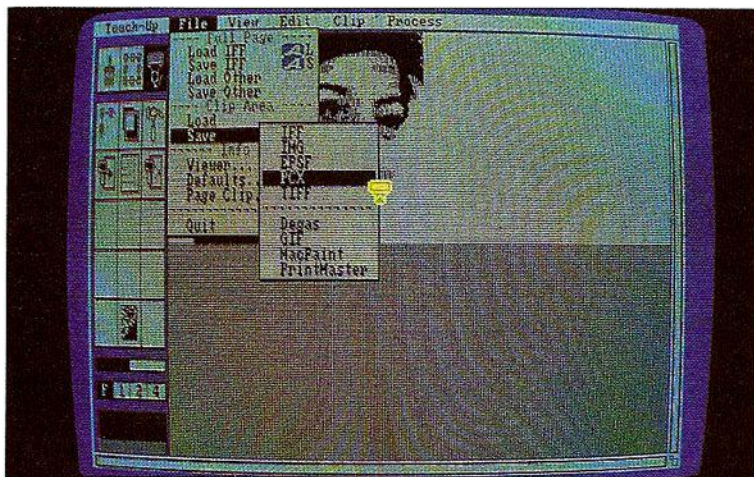
Di solito apparecchi del genere sono molto costosi e voluminosi, soprattutto se rivolti al mercato professionale. L'Hand Scanner, in pratica, consente "in piccolo" (ma non troppo) le stesse prestazioni, senza però consentire il trattamento del colore, e grazie ad una banale operazione di "scorrimento" manuale dello strumento sull'immagine da digitalizzare.

L'esclusione del colore, si badi, è però da intendere solo nel senso che non è in grado di riprodurlo, ma può benissimo digitalizzare (per esempio) una foto a colori trasformando le variazioni **cromatiche** in gradazioni di **grigio**.

Nulla vieta comunque, grazie al completo software di gestione che è compreso nel package e ad interventi sull'immagine mediante comuni tools grafici (**DPaint**, **Photon Paint**, eccetera) che il colore venga aggiunto in un secondo tempo.

Ma procediamo con ordine. Lo scanner, da un punto di vista strettamente hardware, va connesso tramite una sua interfaccia specifica alla **porta parallela** di Amiga. Una volta installato, non rimane altro da fare che attivare da Workbench il suo programma di gestione, **Touchup**, debitamente fornito di un esteso manuale di oltre **160 pagine**.

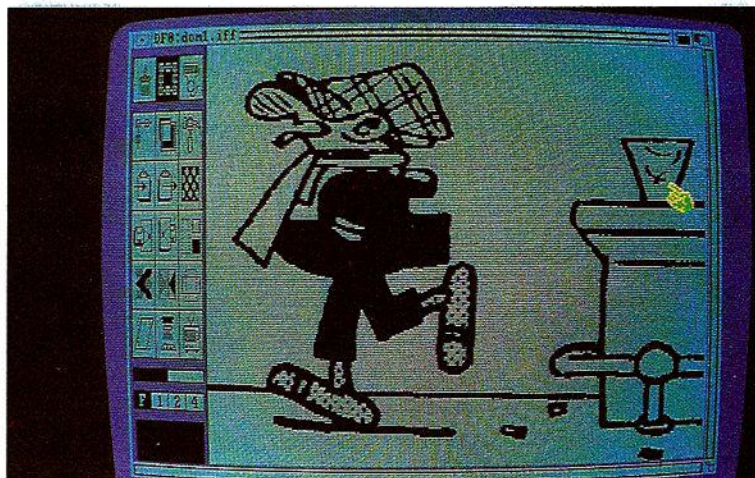
Si tratta, in pratica, di un ambiente integrato dal quale è possibile tanto attivare le operazioni di digitalizzazione vere e proprie, quanto manipolare graficamente e salvare su disco l'immagine ottenuta. Concretamente, la fase di "scannerizzazione" può essere riassunta in poche fasi. Anzitutto va scelta dalla serie di icone grafiche di **Touchup** l'opzione adeguata, guarda caso caratterizzata dal disegno di un piccolo scanner. In questo caso, si otterrà il trasferimento dell'immagine originaria sull'intero schermo. In alternativa, si può anche delimitare una porzione di schermo nel quale far rientrare l'immagine, creando un classico **clip** dai margini impostabili come più aggrada. E già questo non è niente male. Per la fase manuale, è sufficiente impugnare lo scanner, che nel frattempo avrà acceso automaticamente una lampada sulla sua superficie inferiore, e farlo scorrere sull'immagine mantenendo pressato il pulsante **Start** presente sulla sinistra dello strumento.

Le prime volte occorrerà sicuramente qualche prova... di assestamento per

abituarsi alla corretta velocità di scorrimento, ma soprattutto per settare correttamente tutti gli standard modificabili. L'apparecchio è infatti dotato di **tre selettori** da regolare prima della digitalizzazione, e i cui effetti sono tra di loro cumulabili.

In particolare, con una manopola sarà impostabile il **contrasto** che si intende dare all'immagine, con un altro interruttore a 4 posizioni si deciderà sulla **risoluzione** (da 100 a 400 Dpi), ed infine un'altra scelta può riguardare il "modo", ovvero se trattare dei **testi** o delle **foto**. Inutile aggiungere che anche i testi saranno riprodotti nel contesto di una schermata, non certo estraibili come singoli caratteri Ascii, a meno di ricorrere a software **OCR**.

Ultimata questa fase, l'immagine apparirà sullo schermo in pochi secondi. Va detto che la riproduzione è di notevole fattura, pur se limitata al bianco e nero. La resa migliore la si ottiene come ovvio da originali in bianco e nero, ma anche con matrici a colori, Hand Scanner non sfigura, come si può notare dalle riproduzioni mostrate in queste pagine (che, si badi, perdono notevolmente di definizione nella stampa tipografica).

Tutto qua?

Praticamente sì, ma il bello comincia adesso. Clickando sull'icona di **Touchup** occupata da un "pennellone" (di baudiana memoria?), si ha accesso alla sezione grafica vera e propria, che consente già un discreto numero di manipolazioni a carico dell'immagine.

La più scontata, quella di intervenire direttamente su di essa: disegno a **mano libera**, uso di **pennelli** di spessore variabile o addirittura di "**brush**" sullo stile dei più noti tools grafici, e varia altra utilità assortita.

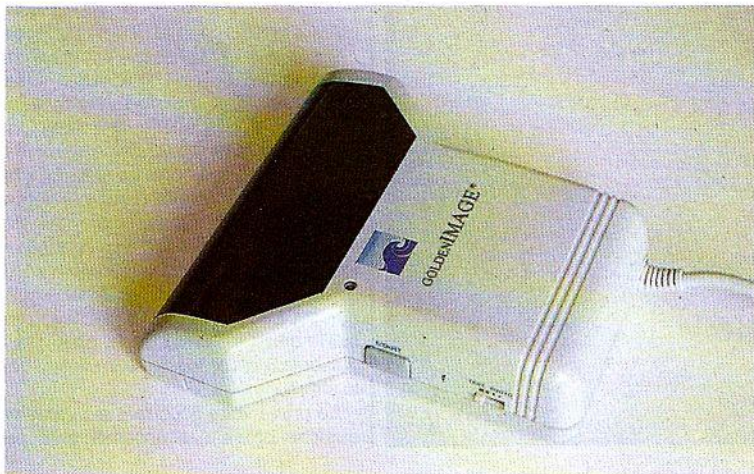
Non manca la possibilità di **invertire il bicromatismo** dello schermo, adoperare un gran numero di **retini** per eventuali riempimenti (**fill**), o ancora **sgranare** l'immagine anche in sue singole sezioni (**dithering**). Molto utili lo **Zoom**, per interventi a livello di pixel, e naturalmente tutti gli automatismi consentiti per il tracciamento di **linee curve**, **poligoni**, **cerchi**, e così via painteggiando.

Di tutto rispetto, poi, la capacità del programma di gestire modificazioni prospettiche dei brush, nonché di importare dall'esterno altri "**clip**", posizionabili in punti prestabiliti dello schermo.

Nulla, insomma, che possa far rimpiangere l'uso di più quotati programmi grafici, che comunque possono anche trovare il loro impiego. Una volta ultimati i lavori su una schermata, questa può infatti essere salvata su disco in formato **IFF**, standard accettato universalmente in ambiente Amiga, pronta per essere ritoccata o solo visualizzata da programmi adeguati. Ma, attenzione, Touchup in questo settore non ha rivali.

Lo schermo, o parte di esso, può essere memorizzato non solo in formato IFF, ma in uno qualunque degli standard correnti, ivi compresi quelli più in "auge" nel mondo degli Ibm compatibili: **Pcx**, **Tiff**, **EpsF**, **Gif**, **Degas**, e addirittura in alcuni formati specifici dell'ambiente Macintosh (**MacPaint**) o di applicativi (**PrintMaster**).

Niente di più facile, insomma, che creare un disco personalizzato da regalare alla propria "ganza", con un suo ritratto che fa da sfondo alle schermate iniziali del boot.

E tutto grazie ad un "piccolo" Hand Scanner.

a cura di Luigi Callegari

AMIGAZZETTA 10

Dopo un periodo di "torpore" dovuto all'assestamento interno della nostra Testata, torniamo a riproporre AmiGazzetta

Amigazzetta è una pubblicazione su supporto magnetico, per Amiga, contenente il meglio del software di pubblico dominio nazionale ed internazionale.

Il dischetto può essere acquistato solamente per corrispondenza, inviando la somma di lire 12.000 a mezzo C/C postale (non si spedisce per contrassegno) all'indirizzo della Systems Editoriale,con la chiara richiesta del numero desiderato.

Su AmiGazzetta pubblicheremo, a cadenza mensile o bimestrale, i migliori programmi di pubblico dominio (= **shareware**) raccolti dalle numerose serie su disco corcolanti per il mondo (Fish Disk, TBAG, Faug, Amiga World Tool Chest, Format, Jumpdisk, Amicus, Slipped...). Accetteremo, ovviamente, anche la collaborazione dei nostri lettori, alle condizioni riportate nel box, per consentire a tutti di fare conoscere i propri lavori, anche se consistono in lunghi listati, altrimenti non pubblicabili su C.C.C. Allo stesso modo, cercheremo di pubblicare da ora in avanti su AmiGazzetta tuttti i **listati Amiga riportati sulla rivista**, per evitare lunghe digitazioni (ed errori) ai nostri lettori. In questo modo potremo trattare programmi molto interessanti, introvabili commercialmente o con caratteristiche tranquillamente paragonabili a pacchetti commerciali, pur essendo del tutto gratuiti.

AmiGazzetta fornirà dunque una quantità di listati **C, Assembler, Basic** (possibilmente tradotti), icone, grafici, programmi di utilità e giocosi, fonti di caratteri, brani musicali, dimostrativi grafici, documentazioni di programmi di pubblico dominio tradotte e quant'altro possano suggerire le richieste dei nostri lettori o la nostra fantasia.

Vediamo sommariamente le descrizioni di alcuni dei programmi inseriti su AmiGazzetta di questo mese (**numero 10**).

Compressori

Utilizzando le banche dati ed i modem, o comunque facendo parte del vasto mondo degli utenti di materiale di pubblico dominio, ci si ritrova invariabilmente a utilizzare i programmi di archiviazione e compressione. Si tratta di programmi dei quali abbiamo già accennato in un articolo apparso sul numero **74** di C.C.C. in grado di ridurre la lunghezza di file su disco e per ridurne i tempi di trasmissione via modem. Gli stessi programmi sono anche in grado di effettuare l'operazione contraria, ritrasformando un file compresso nel gruppo di file (e directory) originari, pronti per essere eseguiti, letti, visualizzati, eccetera.

Il formato dei files compressi è, con pochissime limitazioni (in particolare sulla lunghezza dei nomi), compatibile tra gli stessi programmi per computer diversi. Ciò significa che un file compresso su Ms - Dos ed inviato via modem o trasferito da un dischetto in formato diverso (Ms - Dos, Atari ST...), può essere riconvertito dallo stesso programma di un computer diverso (Amiga, ad esempio).

In AmiGazzetta 10 abbiamo inserito **tutti i compressori / archiviatori** di larga diffusione disponibili per Amiga, certi

di avere reso un grande favore a tutti, che da ora in avanti potranno facilmente maneggiare file inseriti sulle banche dati o distribuiti in collezioni di pubblico dominio. Le versioni fornite sono ovviamente le ultimissime in circolazione, complete dei file di documentazione originali, come vogliono gli autori: **LHarc (1.30), Arc (5.00), Zoo (2.00) e PKAZip (1.01)**. I file di archivio compressi prodotti da questi file si possono riconoscere dal suffisso a tre lettere nel nominativo, rispettivamente: **LZH, ARC, ZPP e ZIP**.

Contenuti

SysInfo (1.91) è un programma che visualizza informazioni complete sull'hardware del proprio computer, comprese le versioni dei chip custom presenti, delle Rom e delle librerie di sistema, eseguendo anche una serie di benchmark comparativi rispetto a versioni "accelerate" di Amiga ed a IBM-compatibili.

Formatter è un elegante programma che consente di formattare e verificare con maggiore velocità ed efficienza i dischetti di AmigaDOS, direttamente dal Workbench, con gadget di controllo secondo il migliore stile Intuition.

VirusX (4.01) è l'ultima versione di un completo ed efficientissimo antivirus, in grado di riconoscere non solo i virus presenti sui bootblock dei dischetti, ma anche quelli residenti in memoria e le forme "mutanti". Per altro, fornisce nella barra del Workbench un orologio con indicazione della quantità di memoria libera costantemente aggiornato, verificando automaticamente l'integrità dei vettori in memoria e i file sui dischetti inseriti.

ChartMaster consente di trasformare serie di punti in grafici di vario tipo: a barre, a torta, ad istogrammi, con possibilità di scalatura, inserimento di testo, variazione dei colori e risoluzione grafica. Uno strumento utilissimo per chi vuole produrre rappresentazioni visive di dati, per poi eventualmente stamparle, magari per evitare l'uso di carta millimetrata e righello per fare contento il professore.

In una apposita directory abbiamo inserito una serie di **icone** adatte per qualunque tipo di uso. Il programma **IconEditor** fornito a corredo consente di manipolare e disegnare icone con la potenza e la flessibilità di un programma grafico alla Deluxe Paint.

Al momento di chiudere C.C.C. stiamo ancora raccogliendo e selezionando programmi recentissimi, tra le dozzine di dischi in nostro possesso, da inserire su AmiGazzetta, che comprenderà comunque anche un completo indice di tutti i programmi apparsi sui numeri precedenti.

Si rammenti che i file destinati su AmiGazzetta vengono tutti "compressi" tramite appositi programmi (restando eseguibili direttamente da Workbench o Shell) in modo da poterne inserire quanti più possibile. In pratica, su 880 K di dischetto sono inseriti mediamente almeno **1 Megabyte di software** di pubblico dominio, completo di documentazioni.

Come collaborare con AmiGazzetta

Sulle pagine di Commodore Computer Club non possono essere pubblicati listati troppo lunghi, ma sappiamo che molti nostri lettori hanno lavori di questo tipo che sarebbero ben felici di divulgare tra altri lettori della nostra rivista.

Chi volesse dunque inserire nel *circuito di pubblico dominio* i propri listati o programmi, può inviarli su dischetto in formato AmigaDOS alla Redazione, accompagnati da una dichiarazione scritta e firmata dove si specifica che il materiale è inviato liberamente e gratuitamente per la pubblicazione su AmiGazzetta, senza richiesta di compensi.

Il materiale deve ovviamente essere originale, possibilmente accompagnato da un file ASCII puro (senza impaginazione, caratteri di controllo, stili o fonti di caratteri speciali), che ne descriva le prestazioni e l'uso, anche solo brevemente.

Sono particolarmente graditi i programmi corredati dei listati sorgente, meglio se commentati, in qualunque linguaggio: BASIC (AMOS, True Basic, Amos, GFA...), C, Assembler, Modula-2 eccetera.

Non verranno considerati programmi copiati da altri autori, non inviati su dischetto o privi completamente di note d'uso e dell'autorizzazione alla pubblicazione libera e gratuita come materiale di pubblico dominio.

Il materiale sarà pubblicato su AmiGazzetta ad insindacabile giudizio della redazione, nei tempi e nei modi da essa stabiliti, senza corrispondere alcun compenso nè in contanti nè in materiale agli autori, ma inserendo, se richiesto, l'indirizzo dell'autore ed eventuali richieste di contributo volontario (shareware) da parte degli acquirenti.

Saranno anche gradite le traduzioni in italiano di programmi di pubblico dominio di provenienza estera.

MandelVroom

In ogni numero di AmiGazzetta pubblicheremo un programma "speciale", in grado di interessare un gran numero di lettori. Di questo programma forniremo possibilmente le istruzioni completamente tradotte in italiano sulle pagine di C.C.C., per consentire ai lettori di quest'ultima di potere apprezzare le caratteristiche del programma (e, speriamo, di decidere l'acquisto di AmiGazzetta...), ed a chi acquista AmiGazzetta o magari possiede già il programma, di avere preziose istruzioni complete in italiano stampate su carta.

Il programma speciale di AmiGazzetta 10 è **MandelVroom**, sicuramente il più sofisticato generatore di grafici frattalici esistente (non solo nel pubblico dominio) per personal computer, di cui forniamo ora la traduzione del testo originale inglese (presente sul disco in formato ASCII) redatto dall'autore.

Istruzioni

MandelVroom è un generatore di grafici di Mandelbrot che consente di esplorare il cosiddetto insieme di Mandelbrot. Fornisce strumenti per zoomare verso l'interno e l'esterno, nonchè per spostarsi attorno al grafico in elaborazione. Consente di creare **curve di Julia** con maschera basata su di una locazione dell'insieme di Mandelbrot oppure in una curva di Julia precedentemente definita. In questo documento, lo spostamento entro gli insiemi viene definito "navigazione".

Grafici di Mandelbrot

Ciascun pixel in un grafico di Mandelbrot rappresenta una locazione nel piano complesso. Il colore di ciascun pixel è determinato dal numero di volte per cui l'equazione può essere reiterata alla locazione del pixel stesso nel piano complesso. Quando si calcola un grafico di Mandelbrot, il risultato è una matrice bidimensionale di conteggi di iterazioni (un elemento per ogni pixel). Una dimensione della matrice rappresenta la larghezza, l'altra l'altezza della finestra del grafico.

MandelVroom usa tale matrice bidimensionale di altezze come base per creare un oggetto tridimensionale. Immaginando un grafico di Mandelbrot come appiattito sulla base dello schermo, i contatori di iterazione dell'equazione di Mandelbrot rappresentano l'altezza.

Tutti i programmi di Mandelbrot usano il contatore di iterazione (cui faremo riferimento chiamandolo "altitudine") in colore. Molti usano il risultato di altitudine modulo numero di colori per determinare il colore da assegnare ad ogni dato pixel. MandelVroom usa tecniche di "contour mapping" per rendere l'altitudine in un colore e per consentire all'utente un controllo diretto sulla colorazione del disegno.

MandelVroom consente di colorare i disegni di Mandelbrot tramite una chiave di mappatura di contorno con 256 elementi. Ciascun elemento di colore (definito come *contorno* in questa documentazione) rappresenta una gamma di altitudini e di colori associati.

Usando **Contour Palette** si possono scegliere le penne da associare ad una data gamma di altitudini. E' prevista anche una palette di colori per mezzo della quale si scelgono i colori da associare ad un elemento di contorno. Si può fissare una penna di contorno cliccando un gadget di comando e ritoccando poi la palette di colori. Per fissare il limite inferiore della gamma di altezze colorate, si usa l'apposito gadget di comando e poi si clicka in un qualche posto dell'insieme di Mandelbrot raffigurato. Questo schema di colorazione è potente ed immediato.

MandelVroom ricorda l'altezza di ciascun pixel mentre calcola l'insieme, sicchè è possibile, successivamente, colorare il disegno senza doverlo ricalcolare. In questo modo si può davvero risparmiare tempo rispetto ad altri programmi che impiegano ore e giorni per realizzare i disegni. Altro particolare è che MandelVroom crea un **task separato** per generare ciascuno dei grafici di Mandelbrot, in modo da consentire di ricolorare un grafico mentre altri sono in generazione, oppure di fare generare grafici mentre si fa qualcos'altro contemporaneamente con altri programmi. Amiga è proprio unico!

Altre caratteristiche in breve di MandelVroom sono: interfaccia completamente Intuitionizzata con gadget e menu, supporto di tutte le risoluzioni standard di Amiga e del modo EXTRA_HALFBRITE, previsione del modo PAL, supporto dei processori 68020 e 68881, cinque generatori matematici, capacità di sospendere e riprendere a piacere le tracciature, strutturazione a progetto (ogni disegno è un progetto separato, con suoi colori, modi eccetera), I/O dei lavori sulla memoria di massa, multitasking dei progetti, navigazione attraverso i progetti, lente di ingrandimento per aiutarsi nella navigazione, HELP in linea per tutti i gadget ed i menu.

Start up

MandelVroom può essere eseguito su Amiga con 512 K di RAM, ma ovviamente lavora più comodamente con 1 Megabyte. Richiede i seguenti files di libreria nella directory **LIBS:** mathtrans, mathieeedoubbas e icon (se lanciato da Workbench). E' inoltre necessario eseguire

un'assegnazione per indicare al programma dove trovare le directory di lavoro. Il comando...

```
Assign Mandelbrot: dh0:Mandel
```

...ad esempio assegna ad una directory Mandel sul disco rigido.

Per essere lanciato da Shell è necessario uno stack (da assegnare tramite l'apposito comando Stack) di almeno 30 K. La sintassi è:

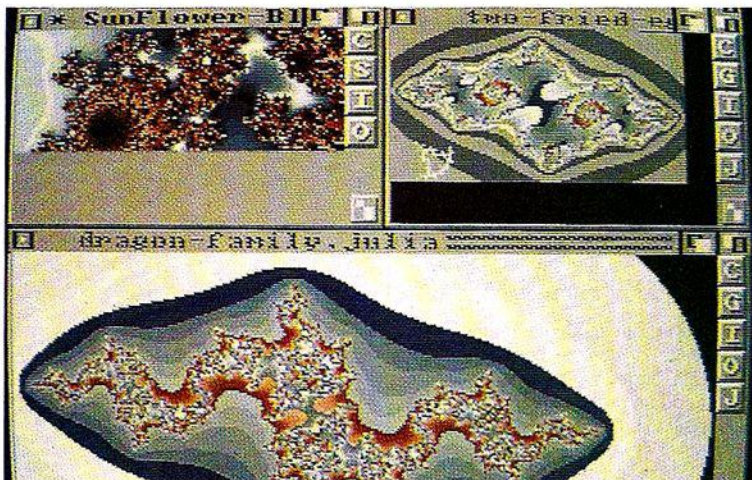
```
MandelVroom <nomeproget> <nome-
proget>...
```

Se si specificano nomi di progetti (grafici precedentemente salvati dal programma) MandelVroom si apre e mostra automaticamente quel progetto in una finestra.

Help

Per non appesantire la documentazione su carta e fornire un aiuto più efficiente e rapido, MandelVroom incorpora files che forniscono informazioni sui propri gadget e menu. Scegliendo da menu Help (menu **Project**), e selezionando col mouse un qualunque gadget o nome di funzione di menu, si otterrà sul video un testo descrittivo.

Picture Window

MandelVroom mostra i progetti in finestre chiamate "picture windows". Ciascun grafico ha i consueti gadget di sistema: chiusura, cambiamento di profondità, trascinamento e ridimensionamento. Sulla destra troviamo invece quattro o cinque gadget usati per controllare MandelVroom:

C - Current

Rende corrente il progetto. E' vitale quando si ha più di un progetto aperto contemporaneamente sullo schermo. Come sempre in Amiga, uno solo è il progetto attivo in un dato momento. Il progetto attivo ha un asterisco nella finestra.

G (o S) - Commutazione

Commuta tra Generate e Stop. E' il bottone che avvia o ferma il generatore grafico. In pratica indica anche lo stato del progetto (in Generazione o Stopped). Può commutare continuamente lo stato di generazione del grafico. Queste funzioni sono disponibili anche nel menu Calculate Generate, che fornisce anche Resume, la quale consente di proseguire la generazione anche dopo un arresto: si può fermarla, salvarla su disco e uscire da MandelVroom e, in seguito, ricaricare il progetto facendolo ripartire da MandelVroom tramite Resume.

I - Zoom

Consente di navigare in regioni già visualizzate.

O - Zoom

Consente di navigare in regioni esterne alla corrente.

J - Fissa seme di Julia

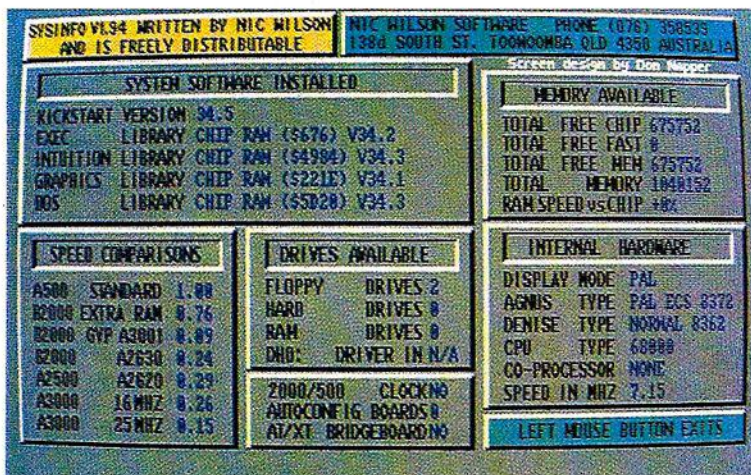
E' presente solo nei grafici di Julia e fissa il seme di generazione.

I menu

MandelVroom è un programma orientato alla grafica di tipo "indica e clicca", quindi controllato da mouse, gadget e menu. I quattro menu sono: Project, Display, Calculate e Special. Il **primo** contiene funzioni che influenzano globalmente un progetto, il **secondo** controlla lo schermo e le finestre di lavoro, il **terzo** regola i modi di calcolo di MandelVroom ed il **quarto** regola i preset e le definizioni di orbite.

Preset

MandelVroom contiene alcuni progetti iniziati che consentono ai dilettanti di sperimentare (la directory è projects). Per aprire uno dei progetti si scegla **Special / Preset Item.** Ogni progetto ha un nome simbolico per mezzo del quale può essere indicato e caricato.

MandelVroom genera il grafico di un progetto non appena è stato caricato, aprendo una finestra ed iniziando i calcoli

e la resa. La finestra è dimensionabile e si possono usare i gadget per sospendere o zoomare il grafico. Usando la funzione **Pan Item** di **Calculate** si può riposizionare il grafico nella finestra ingrandita.

Colori

La Color Palette consente di modificare i valori dei colori dei registri hadrware di Amiga (chiamati anche "penne"). Viene aperta tramite Colors Item / Display. Vi si trovano tre aree separate: al centro tre gadget potenziometrici, uno per ciascuna delle tre componenti cromatiche della penna: rossa, verde e blu (RGB). Sulla destra tre gadget consentono di controllare i colori delle penne, una per volta però, indicata con un click del mouse. I numeri riportati sono in esadecimale ed indicano le quantità di colori principali nella tinta. Sono permesse varie operazioni di manipolazione della tavolozza che chiunque abbia usato programmi tipo Deluxe Paint saprà usare.

Contorni

Un contorno è l'associazione di una penna ad una gamma di altezze del grafico. Per aprire la finestra di Contour si scelga Display Contours: si aprirà una grossa finestra con quattro aree principali: la barra di intestazione (con informazioni sul contorno corrente), i gadget di comando per redigere i contorni, la barra dei colori (penne per 256 contorni, poste sotto i gadget) e una sezione di Contour Editing sotto la barra dei colori. La Bar Colour e la Contour Editing sono correlate: la prima mostra le 256 penne associate ai contorni, la seconda mostra i 32 colori tra i 256 contorni possibili. La barra dei titoli tiene aggiornati sul contorno sul quale stiamo operando: scegliendo un elemento di contorno, ne mostra subito il numero, la penna associata e la gamma di altezze associate. Ad esempio:

C: 1 P:4 H: 1022-137

indica che il contorno corrente è 1, la penna è la 4 e la gamma di altezze rese con questo varia tra 1022 a 137.

I gadget di editing del contorno vengono rappresentati da una colonna che contiene un gadget di penna (in cima) ed un gadget di potenziometro. Il gadget di penna viene usato per scegliere un contorno come corrente, indicato da un riquadro intorno ad esso.

Il potenziometro di contorno serve a due scopi: innanzitutto indica approssimativamente l'altezza del contorno dato (è la barra che dà il valore preciso) e poi consente di fissarne rudimentalmente l'altezza. Il comando Set consente di modificare in modo efficace le altezze, ma è poco efficiente se si vogliono modificare tutte le 256 possibili. Si può allora usare il comando SMOOTH indicando due penne di contorno tra le quali far sfumare automaticamente la tinta al programma.

Ceiling

Il gadget di Ceiling è un potenziometro sulla destra del Contour Editing. Consente di non influire sul disegno, sostanzialmente, in quanto agisce solo sul modo in cui le altezze di contorno vengono mostrate nei gadget dei potenziometri di contorno. E' più utile per visionare quei contorni le cui altezze sono relativamente basse. Se si tira verso il basso il ceiling a 50, ad esempio, si ottiene una buona visione di quelle altezze di contorno che sono di cinquanta o meno.

Paint

Dopo avere modificato i contorni si può visionare l'effetto sul disegno: basta premere il gadget Paint. MandelVroom ritraccerà il disegno usando le nuove regolazioni dei colori. Tutte le volte che si cambiano i contorni (altezze o penne) bisogna premere il gadget di comando Paint per vederne l'effetto.

Cambiare Contour Pen

Il modo più semplice di cambiare le penne di contorno è usare il comando Set. Prima di usare il comando si deve

aprire la Color Palette: si scelga Edit Colors e si clicki sulla penna della Contour Palette da rendere corrente, poi si clicki il gadget di SET e infine la penna della palette che si vuole usare. Si noti che la barra dei colori cambia per indicare la nuova penna.

Autocontour

I comandi Autocontour sono usati per fissare le altezze del disegno corrente in modo da distribuirle uniformemente sui contorni. Si possono scegliere le maschere delle penne, le lunghezze delle maschere ed i colori delle penne. Autocontour sceglierà da solo le altezze dei contorni: ovviamente le lunghezze scelte per le maschere (pattern) influiscono sulla nitidezza del disegno. Quando si sceglie Autocontour I, MandelVroom esegue i successivi passi: conta il numero di contorni nella palette relativa ed iniziando con il primo contorno associato ad una penna differente dalla 1. Calcola poi il numero di pixel del disegno meno il numero di pixel con la stessa altezza di MaxIteration e divide il numero di pixel così ricavati col numero di contorni calcolati precedentemente. Infine calcola anche un istogramma del numero di pixel a ciascuna delle possibili altezze.

Ciclaggio colori

Scegliendo Display / Cycling si accede al pannello di controllo del ciclaggio dei colori. Questo è un metodo di animazione che tratta una gamma di penne consecutive nella palette di colori come un registro a rotazione. Quando "cicla", il programma muove i componenti cromatici di una penna in un'altra penna. Ovviamente una gamma di penne consecutive viene scandita secondo una direzione ed una velocità di rotazione. Per usare la palette di color cycling bisogna avere aperto anche la Color Palette: si ottiene ciò semplicemente scegliendo Display dal menu Colors. Si possono scegliere sino a quattro gamme di rotazione dei colori. Si sceglie prima la penna, poi si clicka sul gadget di RANGE e poi ancora su di un'altra penna per indicare la gamma di scorrimento. I gadget C1, C2, C3 e C4 indicano il numero di gamma di ciclaggio, con l'attuale evidenziato. La direzione di scorrimento viene invertita tramite il gadget DIRECTION. Il gadget SPEED consente di regolare linearmente la velocità di rotazione.

PANorama

Il comando PAN consente di fare scorrere il progetto nel piano complesso. Per usarlo si deve scegliere il sottomenu PAN, porre il cursore nel progetto corrente, premere e mantenere premuto il tasto sinistro del mouse, spostare il mouse per indicare, rilasciare il pulsante sinistro e scegliere PAN Generate per fare ritracciare il disegno.

Max Iteration

MandelVroom consente di stabilire il massimo numero di iterazioni tramite Max Iteration del menu Calculate. Il numero intero specificato nel requester aperto da questa funzione indica il massimo numero di ripetizioni del calcolo eseguito dal programma sull'equazione di partenza.

Questa è una limitazione necessaria, perchè alcuni punti provocherebbero una iterazione infinita, se non si indicasse un numero massimo di ripetizioni, per la natura stessa dell'equazione. Si badi bene dal settare al massimo valore (1023) il parametro, perchè se i disegni diventano molto accurati, se contiene molti punti di iterazione lunga la resa grafica potrebbe richiedere moltissimo tempo di generazione.

Orbit

Un'orbita, nel contesto dell'insieme di Mandelbrot e delle curve di Julia, è la sequenza di tutte le curve calcolate in ciascun passo del processo di generazione (sino al valore scelto per Max Iteration). La finestra di orbita indica una mappa della sequenza nel piano complesso. Per osservare un orbitale bisogna aprire la finestra di Orbit usando la funzione di menu Special / Orbit: la finestra di orbita mostrerà una mappa della sequenza nel piano complesso. Quando si preme il pulsante sinistro del mouse, MandelVroom traccia il disegno di una orbita basata sul pixel che si preleva dal disegno di Mandelbrot o di Julia. Facendo scorrere il mouse intorno mentre si tiene premuto il pulsante sinistro del mouse, si modifica la visualizzazione dell'orbita concordemente. I due parametri che influiscono sulla visualizzazione dell'orbitale sono Maximum Orbit Iteration e Math Mode.

Matematica

MandelVroom prevede tre formati di dati numerici. Il più veloce e preciso è un formato intero frazionale usato per tutti i preset. Il successivo, molto più lento, è il formato Fast Floating Point di Amiga: ambedue i formati, intero e FFP, usano variabili a 32 bit, ma FFP è più preciso perchè usa la virgola mobile. L'ultimo formato è lo IEEE a 64 bit in virgola mobile, che è il più preciso dei tre ma, se non si dispone di hardware supplementare per i calcoli in virgola mobile (coprocessore matematico) è sconsigliabile perchè estremamente lento. In ogni caso consente di zoomare molto più profondamente degli altri formati.

La funzione Math Mode ha cinque sottoelementi:

68000 - Interi a 32 bit e moltiplica a 16

FFP - Fast Floating Point Amiga

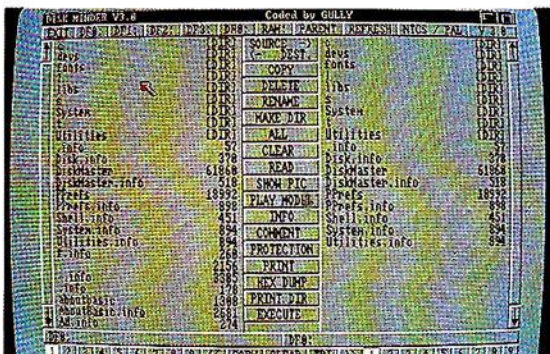
IEEE - Librerie di sistema f. p. a 64 bit

68020 - Librerie a 32 bit con moltiplica a 32

68881 - IEEE a 64 bit f. p. con FPU 68881/2

Si noti che il modo 68881 non lavora attualmente con schede acceleratrici tipo CMI per incompatibilità di progetto.

I/O di progetti

I progetti generati da MandelVroom possono essere registrati su disco e ricaricati successivamente per ulteriori elaborazioni. Save registra su disco il progetto attualmente selezionato, specificando il nome tramite un requester. Load esegue il caricamento in modo simile. Per salvare schermate come files ILBM si usi l'apposita funzione del menu Save ILBM in Project.

di Giancarlo Mariani

PASSEGGIANDO SULLO SCHERMO

Lo schermo di un computer Ms-Dos può essere "trattato" come eravamo abituati a fare con il C/64

Molti degli attuali utenti Ms - Dos provengono da sistemi più piccoli ed antiquati (leggi Vic/20, C/64). Sono ormai più che note le traversie alle quali erano sottoposti gli antichi utilizzatori di queste macchine, costretti a servirsi del modesto Basic 2.0 che NON possiede una miriade di istruzioni utilissime.

In particolare, si notava la mancanza di istruzioni per la gestione del video. Qualsiasi operazione ad esso diretta (a parte PRINT che, per ovvie ragioni, era inclusa nel Basic...) presumeva il ricorso a svariate (e numerosissime) POKE.

Come eravamo

Come tutti gli smanettoni ricorderanno, tali computer possedevano due distinte aree di memoria; una, detta memoria di schermo, per contenere i **caratteri** da visualizzare; l'altra, detta memoria **colore**, per contenere i colori ad essi relativi.

Prendiamo il caso specifico del C/64: la parte di memoria compresa tra la locazione **1024** alla **2023** è destinata alla memoria di schermo e risulta organizzata in 25 righe da 40 caratteri ciascuno per un totale di 1000 caratteri, quanti sono quelli effettivamente visualizzabili sullo schermo.

POKando un valore compreso tra **0** e **255** in una qualsiasi cella video, si otteneva la visualizzazione del carattere corrispondente. Ad esempio, con **POKE 1024, 65** si visualizzava una "A" nell'angolo in alto a sinistra dello schermo. Questo però SOLO a patto che il carattere visualizzato venisse anche colorato.

L'area di memoria lunga 1000 bytes, che parte dalla locazione **55296** e termina alla **56295**, risulta organizzata proprio come la memoria di schermo ma si riferisce al colore di ciascun carattere.

Per colorare la "A" visualizzata prima, occorre anche impartire **POKE 55296, Colore**. La tecnica descritta veniva usata sui Vic/20 e C/64 prevalentemente per simulare istruzioni PRINT in posizioni determinate di schermo (tipo **Print at**, istruzione assente nel Basic 2.0). Dalla riga e colonna di visualizzazione si determinava l'indirizzo di memoria di schermo e di memoria colore corrispondente e quindi si scriveva, tramite Poke, la stringa desiderata.

Come siamo

Nel mondo IBM non è più necessario ricorrere a questi "trucchi" dal momento che tutti i linguaggi evoluti, sviluppati sotto Dos, possiedono istruzioni specifiche per posizionare il cursore, per cambiare colore ai caratteri, e così via (come, ad esempio, **LOCATE** e **COLOR** del Gw-Basic).

A livello di conoscenza, comunque, può essere utile sapere come il PC organizza la sua memoria video e come fare per sfruttarla.

Anche il PC, come il C/64, possiede un'area di memoria interamente dedicata al video in modo testo. Questa, in caso di scheda video **a colori** (da **CGA** in poi) è lunga **4000** bytes, 2000 dei quali dedicati ai **caratteri** (80 colonne x 25 righe) e gli altri 2000 ai **colori** corrispondenti.

In pratica, l'area contiene l'immagine video (in modo testo, badate bene) così come si vede sullo schermo. La memoria di schermo del PC è organizzata in modo leggermente diverso da quello del C/64. I suoi 4000 bytes sono organizzati in 25 righe da 160 bytes ciascuna.

Di questi, 80 saranno dedicati alla memorizzazione dei caratteri, mentre i rimanenti 80 ai colori corrispondenti. Il metodo di memorizzazione è il seguente:

```
0 = Nero            8 = Grigio scuro
1 = Blu             9 = Blu chiaro
2 = Verde          10 = Verde chiaro
3 = Azzurro        11 = Azzurro chiaro
4 = Rosso          12 = Rosso chiaro
5 = Viola          13 = Viola chiaro
6 = Marrone        14 = Giallo
7 = Grigio chiaro  15 = Bianco


Byte + 0     67     (Codice ASCII "C")
Byte + 1     15     (Codice colore Bianco)
Byte + 2     73     (Codice ASCII "I")
Byte + 3     15     (Codice colore Bianco)
Byte + 4     65     (Codice ASCII "A")
Byte + 5     15     (Codice colore Bianco)
Byte + 6     79     (Codice ASCII "O")
Byte + 7     15     (Codice colore Bianco)
```

Codice dei colori per lo schermo di un Ms - Dos compatibile. Ad esempio, scrivendo "CIAO" in alto a sinistra dello schermo in colore bianco, i primi 8 bytes della memoria video saranno composti come in tabella.

```
10 REM SAVESCRN.BAS :
15 REM   Salvataggio schermata modo testo 80x25
20 REM   Scheda a colori CGA o Seguenti
25 REM   Versione GWbasic e QuickBasic
30 REM
100 OPEN "NomeFile" FOR OUTPUT AS #1
110 DEF SEG = &HB800
120 FOR Riga = 0 TO 24
130 FOR Colonna = 0 TO 79
140  IndirCaratt = (Riga * 160) + (Colonna * 2)
150  IndirColore = IndirCaratt + 1
160  Caratt% = PEEK(IndirCaratt)
170  Colore% = PEEK(IndirColore)
180  PRINT #1, CHR$(Caratt%);
185  PRINT #1, CHR$(Colore%);
190   NEXT Colonna
200 NEXT Riga
210 CLOSE #1
220 DEF SEG
230 END
```

```
10 REM LOADSCRN.BAS :
15 REM Carica schermo
17 REM modo testo 80x25
20 REM Scheda CGA o Seguenti
25 REM GWbasic e QuickBasic
30 REM
100 OPEN"NomeFile" FOR INPUT AS #1
110 DEF SEG = &HB800
120 FOR Riga = 0 TO 24
130 FOR Colonna = 0 TO 79
140 IndirCaratt = (Riga * 160) + (Colonna * 2)
150 IndirColore = IndirCaratt + 1
160 Caratt$=INPUT$(1,#1) : Colore$=INPUT$(1,#1)
170 POKE IndirCaratt,ASC(Caratt$)
180 POKE IndirColore,ASC(Colore$)
190 NEXT Colonna
200 NEXT Riga
210 CLOSE #1
220 DEF SEG
230 END
```

Nel **primo** byte della memoria è presente il codice del carattere posto in alto a sinistra dello schermo (posizione 0, 0). Nel **secondo** byte è memorizzato il colore del carattere. Nel terzo byte c'è il codice del carattere posto nella posizione successiva (posizione 0, 1); nel quarto il colore ad esso relativo. E così via, fino alla fine. In pratica, vengono memorizzati in modo alternato carattere0 - colore0, carattere1 - colore1, ... carattere2000 - colore2000.

Sia i caratteri che i colori sono ovviamente codificati: i primi secondo il codice ASCII (esaminare un qualsiasi manuale per i dettagli); i colori secondo la tabella riportata in basso.

In realtà i 2000 bytes dedicati ai colori dei caratteri determinano anche il **colore di fondo** che deve possedere il carattere stesso. Il colore di fondo è determinato dai 4 bit più alti del byte del colore, mentre il colore del carattere dai 4 bit più bassi.

Uno qualsiasi dei 2000 bytes dedicati al colore è così formato:

```
Bit 7  6  5  4  3  2  1  0
    L  S  S  S  C  C  C  C
```

I bit contrassegnati con "**C**" (bit 0... bit 3) determinano il colore del **carattere**; quelli con "**S**" (bit 4... bit 6) determinano il colore di **fondo** del carattere, secondo i primi 8 colori della tabella (da 0 = nero a 7 = grigio chiaro); il bit 7, contrassegnato con **L**, determina se il carattere deve **lampeggiare** (bit settato, = 1) oppure no (bit resettato, = 0).

Per determinare il valore da POKare nella memoria colore, il calcolo è il seguente:

Valore = ColCarat + 16 * ColSfondo + 128 * Lampeggio

```
10 REM SAVESC_2.BAS :
15 REM Salvataggio scherm. modo testo 80x25 con SCREEN
20 REM   Tutte le schede video
25 REM   Versione QuickBasic e GWbasic
26 REM   Per GWbasic, riga 120: FOR 1 To 24
30 REM


100 OPEN "NomeFile" FOR OUTPUT AS #1
120 FOR Riga = 1 TO 25
130 FOR Colonna = 1 TO 80
140  Caratt% = SCREEN(Riga, Colonna, 0)
150  Colore% = SCREEN(Riga, Colonna, 1)
180  PRINT #1, CHR$(Caratt%); : PRINT #1, CHR$(Colore%);
190 NEXT Colonna
200 NEXT Riga
210 CLOSE #1
230 END
```

```
10 REM BLOAD.BAS
20 REM Caric. scher. BLOAD
30 REM GWbasic e QuickBasic
40 REM
100 DEF SEG = &hB800
110 BLOAD "NomeFile",0
120 DEF SEG
130 END
```

```
10 REM BSAVE.BAS
20 REM Salva schermo con BSAVE
30 REM GWbasic e QuickBasic
40 REM
100 DEF SEG = &hB800
110 BSAVE "NomeFile",0,4000
120 DEF SEG
130 END
```

```
/* SAVESCRN.C */
/* Salva schermo modo testo 80x25*/
/* Scheda a colori CGA o Seguenti */
/* Versione TurboC */

#include<io.h>
#include<dos.h>


void main()
{
  unsigned char Caratt[1],Colore[1];
  int       PicFile;
  int       Riga,Colonna;
  int       IndirCaratt, IndirColore;

  PicFile = _creat("NomeFile",FA_ARCH);
  for (Riga=0; Riga<=24; Riga++)
  {
    for (Colonna=0; Colonna<=79; Colonna++)
    {
    IndirCaratt = (Riga * 160)+(Colonna * 2);
      IndirColore = IndirCaratt + 1;
      Caratt[0] = peekb(0xB800,IndirCaratt);
      Colore[0] = peekb(0xB800,IndirColore);
      write(PicFile,Caratt,1);
      write(PicFile,Colore,1);
    }
  }
  close (PicFile);
}
```

```
/* LOADSCRN.C :                                     */
/* Caricamento schermata in modo testo 80x25 */
/* Scheda a colori CGA o Seguenti                   */
/* Versione TurboC                                  */

#include<io.h>
#include<dos.h>
#include<fcntl.h>

void main()
{
  unsigned char Caratt[1],Colore[1];
  int             PicFile;
  int             Riga,Colonna;
  int             IndirCaratt, IndirColore;

  PicFile = open("NomeFile",O_BINARY);
  for (Riga=0; Riga<=24; Riga++)
  {
    for (Colonna=0; Colonna<=79; Colonna++)
    {
      IndirCaratt = (Riga * 160) + (Colonna * 2);
      IndirColore = IndirCaratt + 1;
      read(PicFile,Caratt,1);
      read(PicFile,Colore,1);
      pokeb(0xB800,IndirCaratt,Caratt[0]);
      pokeb(0xB800,IndirColore,Colore[0]);
    }
  }
  close (PicFile);
}
```

...in cui **ColCarat** è il colore del carattere (variabile tra 0 e 15); **ColSfondo** è il colore dello sfondo (da 0 a 7) e **Lampeggio** è il flag per il lampeggio (0 oppure 1).

```
/* GETTEXT.C : Salvataggio */
/* schermata modo testo    */
/* 80x25 tramite GETTEXT   */
/* Scheda CGA o Seguenti   */
/* Versione TurboC         */

#include <dos.h>
#include <conio.h>
#include <io.h>
void main ()
{
 unsigned char Schermo[4020];
  int           PicFile;

gettext(1,1,80,25,Schermo);
PicFile=_creat("NomeFile",FA_ARCH);
  write(PicFile,Schermo,4000);
  close(PicFile);
}
```

Anche sui PC, quindi, mediante semplici POKE riusciremo a scrivere ed a colorare il nostro video. Ma DOVE è posta la memoria di schermo?

Spazio video

Dal momento che il processore può "vedere" direttamente una limitata quantità di memoria (fino ad 1 M Bytes in modo normale, altrimenti parecchia di più), questa viene divisa in blocchi lunghi ciascuno 64 k, detti **Segmenti**.

Al video è stato riservato uno di questi segmenti, e precisamente, sempre per

```
/* PUTTEXT.C :                   */
/* Caricam. schermata in modo testo */
/* 80x25 con istruzione PUTTEXT  */
/* Scheda a colori CGA o Seguenti */
/* Versione TurboC               */


#include <io.h>
#include <conio.h>
#include <fcntl.h>


void main ()
{
  char          Schermo[4096];
  int           PicFile;
  int           k;

  PicFile = open("NomeFile",O_BINARY);
  read(PicFile,Schermo,4000);
  puttext(1,1,80,25,Schermo);
  close (PicFile);
}
```

```
{ SAVESCRN.PAS :                                   }
{ Salvataggio schermata in modo testo 80x25 }
{ Scheda a colori CGA o Seguenti            }
{ Versione TurboPascal                      }

uses Crt;

type Car = array[0..0] of byte;

var  Caratt,Colore            : Car;
     PicFile                  : file of Car;
     Riga,Colonna             : integer;
     IndirCaratt, IndirColore : integer;

begin
  assign(PicFile,'NomeFile');
  rewrite(PicFile);
  for Riga:=0 to 24 do
  begin
    for Colonna:=0 to 79 do
    begin
      IndirCaratt := (Riga * 160) + (Colonna *
2);
      IndirColore := IndirCaratt + 1;
      Caratt[0] := Mem[47104:IndirCaratt];
      Colore[0] := Mem[47104:IndirColore];
      write(PicFile,Caratt);
      write(PicFile,Colore);
    end;
  end;
  close (PicFile);
end.
```

```
{ LOADSCRN.PAS :                                   }
{ Caricamento schermata in modo testo 80x25 }
{ Scheda a colori CGA o Seguenti            }
{ Versione TurboPascal                      }

uses Crt;
type Car = array[0..0] of byte;

var  Caratt,Colore            : Car;
     PicFile                  : file of Car;
     Riga,Colonna             : integer;
     IndirCaratt, IndirColore : integer;

begin
  assign(PicFile,'NomeFile');
  reset(PicFile);
  for Riga:=0 to 24 do
  begin
    for Colonna:=0 to 79 do
    begin
IndirCaratt := (Riga * 160) + (Colonna * 2);
      IndirColore := IndirCaratt + 1;
      read(PicFile,Caratt);
      read(PicFile,Colore);
      Mem[47104:IndirCaratt]:=Caratt[0];
      Mem[47104:IndirColore]:=Colore[0];
    end;
  end;
  close (PicFile);
end.
```

quanto riguarda la scheda a colori (**CGA**, **EGA** oppure **VGA**) viene posto a partire dall'indirizzo esadecimale **$B800** (corrispondente al decimale **47104**).

I 4000 bytes della memoria di schermo (caratteri + colore) sono posti a partire dalla locazione **0** relativa al segmentó B800. Pertanto:

B800 : 0 conterrà il codice ASCII del carattere posto alla prima locazione di schermo.

B800 : 1 conterrà il colore relativo al carattere precedente.

...eccetera. In pratica:

```
B800: 0 Carattere posizione 0, 0
B800: 1 Col. car. in posiz. 0, 0
B800: 2 Carattere posizione 0, 1
B800: 3 Colore in posizione 0, 1
..........................
B800: 3998 Car. posiz. 24, 79
B800: 3999 Col. posiz. 24, 79
```

Pokando valori opportuni in queste locazioni, riusciremo a scrivere e colorare a piacimento lo schermo del nostro PC in modo testo.

```
{ DETECT.C : Determina automaticamente la scheda grafica }
{            presente nel computer. Turbo-C              }

#include <graphics.h>
#include <stdio.h>

int g_driver,g_mode;
void main()
{
  detectgraph (&g_driver,&g_mode);
  printf ("Il numero della tua scheda e' : %d\n",g_driver);
}
```

Tradurre "Poke"

Il Basic 2.0 del C/64 possiede la famigerata istruzione Poke, che permette di scrivere a piacimento nella memoria del computer. Nei linguaggi dei PC, l'istruzio-

ne è rimasta fondamentalmente la stessa. Vediamola nelle varie versioni:

Gwbasic e QuickBasic

```
POKE Indirizzo, Valore
```

L'**Indirizzo** è quello in cui vogliamo memorizzare il **Valore**.

Come detto prima, però, la memoria del PC è organizzata in **segmenti**; non è quindi sufficiente specificare solo l'indirizzo, ma è necessario anche indicare con precisione il segmento nel quale è posto l'indirizzo desiderato. Ciò si realizza tramite un'altra istruzione Basic...

```
DEF SEG = Indirizzo Segmento
```

...in cui **Indirizzo Segmento** è l'indirizzo iniziale del segmento desiderato. Per scrivere una "A" colorata di bianco nella prima posizione in alto a sinistra dello schermo scriveremo...

```
10 DEF SEG = &hB800: REM Seleziona il segmento video
20 POKE 0, 65: REM Scrive nella prima locazione della memoria video il codice della "A"
30 POKE 1, 15: REM Colora la "A" di bianco
40 DEF SEG: REM Ripristina il segmento originale
50 END
```

L'istruzione posta nella riga 40 (DEF SEG senza parametri) serve a ripristinare il segmento di default del Basic, ossia quello nel quale l'interprete risulta attivo, lavora, gestisce le variabili, e così via.

E' molto importante che quest'ultima istruzione (di "ripristino") sia sempre presente, altrimenti il Basic resta "posizionato" sul segmento video e possono accadere cose imprevedibili.

Per la stessa ragione, l'istruzione DEF SEG andrebbe sempre scritta in un programma e MAI impartita in modo diretto. Inoltre deve essere ben nota la funzione svolta dalle varie locazioni (specificate tramite l'indirizzo in DEF SEG e nelle eventuali istruzioni POKE), perché alterazioni di locazioni casuali di memoria portano quasi sicuramente all'inchiodamento del computer, con relativa necessità di Reset e perdita del programma in memoria.

Quick C

Il C dispone di un'istruzione che esegue sia il cambio di segmento che la scrittura del byte. L'istruzione in oggetto è **Pokeb:**

```
pokeb (Segm, Indirizzo, Valore);
```

...in cui **Segm** è l'indirizzo iniziale del segmento da prendere in considerazione, **Indirizzo** è l'indirizzo relativo al segmento nel quale va scritto il **Valore**, appunto, da scrivere. Il programma di prima, scritto in C, diventa:

```
pokeb (0xB800, 0, 65);/* Scrive "A"*/
pokeb (0xB800, 1, 15);/* la colora di bianco */
```

Anche in questo caso, ovviamente, bisogna fare attenzione a non Pokare a caso, per le ragioni viste prima.

Pascal

Il Pascal (**Turbo Borland** oppure **Quick Microsoft**) si differenzia un po' dagli altri due linguaggi, dato che non possiede istruzioni specifiche per scrivere in memoria. Il Pascal, infatti, possiede al suo interno un array di interi, chiamato **Mem**, già definito e contenente tanti elementi quanti sono i bytes della memoria del PC. Ognuno degli elementi è l'immagine del corrispondente byte di memoria. Per scrivere un byte dovremo quindi impartire...

```
Mem [Segm: Indirizzo] := Valore;
```

...in cui **Segm** è l'indirizzo iniziale del segmento da prendere in considerazione; **Indirizzo** è l'indirizzo relativo al segmento nel quale va scritto il **Valore** da memorizzare.

Il programma, in Pascal, diventa:

```
Mem [47104:0] := 65;
{ Scrive la "A"}
Mem [47104:1] := 15;
{ la colora di bianco }
```

L'ora di Peek

Dal momento che la memoria video è, appunto, una memoria(!), possiede la caratteristica di essere scritta, ma anche letta. Possiamo, infatti, esaminare la memoria video per diversi scopi, magari per effettuare un'HardCopy su stampante, o altro. Per leggere in memoria sono necessari comandi "inversi" a quelli prima esaminati.

Leggere in Basic

Anche in questo caso l'istruzione rispecchia quella del Basic 2.0, Peek. La sintassi è...

```
Valore = PEEK (Indirizzo)
```

...in cui **Indirizzo** è l'indirizzo da esaminare, mentre in **Valore** sarà presente quello, appunto, individuato. Nel caso abbiate digitato, e fatto girare, il programma prima proposto, il corrispondente listato per leggere il contenuto della prima locazione di memoria video potrebbe essere il seguente:

```
10 DEF SEG = &hB800: REM Seleziona il segmento video
20 Caratt = PEEK(0): REM Legge il carattere (0,0)
```

```
0 = richiesta di identif.
1 = CGA
2 = MCGA
3 = EGA
4 = EGA 64k
5 = EGA monocromatica
6 = IBM 8514
7 = Hercules
8 = PC AT&T 6300
9 = VGA
10 = Ms - Dos compat. 3270
```

Valore di graphdriver relativo al tipo di scheda presente nel computer

```
{ DETECT.C : Determina}
{ autom_ticamente la scheda}
{ grafica presente}
#include <graphics.h>
#include <stdio.h>

int g_driver,g_mode;

void main()
{
detectgraph(&g_driver,&g_mode);
printf ("Il numero della tua");
printf ("scheda è : %d\n")
printf (g_driver);
}
```

Esame scheda grafica in C

```
{ POKESCRN.PAS: Programma per generare caratteri
casuali sullo schermo  }
{salvare e ricaricare le schermate di testo}
{ Versione Turbo-Pascal 5.5 }

uses Crt; { Libreria di sistema }

{ Variabili }

var
  VideoSeg : word; { Segmento video }
  Scelta   : integer; { Scelta da menu }
  k        : integer; { Cicli FOR      }
  Numero   : integer; { N. car. da visualizz. }
  Addr     : integer; { Indirizzo video}
  Carat, Colore : integer; { Carattere e colore
da visualizzare }
  a        : char; { Tasto premuto }

{ Prototipi di funzioni }
procedure LoadScreen(Segm:word); forward;
procedure SaveScreen(Segm:word); forward;

{ LoadScreen: Carica la scherm. video da disco  }
{   Segm = Segmento video
}
procedure LoadScreen (Segm:word);
type Screen=array[0..4007] of byte;
var  k       : integer;
     PicFile : file of screen;
     Schermo : screen;
begin
  clrscr;
  assign(PicFile,'SCREEN.SCR');
  reset(PicFile);
  read(PicFile,Schermo);
  close(PicFile);
  for k:=7 to 4007 do mem[Segm:k-7]:=Schermo[k];
end; { LoadScreen }

{ SaveScreen: Salva la scherm. video su disco  }
{   Segm = Segmento video
}
procedure SaveScreen (Segm:word);
type Screen=array[0..4007] of byte;
var  k       : integer;
     PicFile : file of screen;
     Schermo : screen;
begin
  for k:=7 to 4007 do Schermo[k]:=mem[Segm:k-7];
  Schermo[0]:=253;Schermo[1]:=0;Schermo [2] :=
184; Schermo[3]:=0;
  Schermo[4]:=0; Schermo[5]:=160; Schermo[6]:=15;
  assign(PicFile,'SCREEN.SCR');
  rewrite(PicFile);
  write(PicFile,Schermo);
  close(PicFile);
end; { SaveScreen }

{ Main program }
begin
  VideoSeg := 47104; { Segmento video &hB800 }
repeat
  clrscr; writeln('  MENU'); writeln;
  writeln('  0 - Fine programma');
  writeln('  1 - Generazione caratteri');
  writeln('  2 - Caricam. scher. "SCREEN.SCR"');
  writeln; write('  Scelta : '); readln(Scelta);

  if Scelta = 1 then
  begin
    randomize; { Inizial. gen. numeri casuali }
    writeln;
    writeln('  Alla fine della generazione della
schermata digitare');
    writeln('  0 - Torna al menu');
    writeln('  1 - Salva la schermata con il no-
me "SCREEN.SCR"');
    writeln; write('  Quanti caratteri devo gene-
rare : ');
    readln(Numero); clrscr;
    for k := 1 to Numero do
    begin
      Carat := random(223)+32;
      Addr := random(4000);
      Colore := random(15)+1;
      if (Addr mod 2) <> 0 THEN Addr := Addr + 1;
      mem[VideoSeg:Addr] := Carat; mem[VideoSeg
:Addr+1] := Colore;
    end;
    repeat
      a := readkey;
    until (a='0') or (a='1');
    if a = '1' then SaveScreen(VideoSeg) ;
  end;

  if Scelta = 2 then
  begin
    LoadScreen(VideoSeg);
    a := readkey;
  end;
until Scelta=0;

end.
```

Versione Pascal del programma dimostrativo "completo"
Si noti la straordinaria rassomiglianza tra i tre linguaggi più diffusi per i computer Ms - Dos compatibili. I lettori possono confrontare tra loro le tre implementazioni allo scopo di individuare le principali peculiarità dei tre linguaggi fondamentali

```
30 Colore = PEEK(1): REM Legge
il colore (0,0)
40 DEF SEG: REM Ripristina il
segmento originale
45 PRINT Caratt, Colore
50 END
```

Nel nostro caso, la variabile **Caratt** conterrà il valore 65; **Colore** il valore 15.

Leggere in C

L'istruzione C che permette di leggere la memoria è Peekb. La sua sintassi è la seguente:

```
Valore = peekb (Segm, Indir);
```

...in cui **Segm** è l'indirizzo iniziale del segmento, **Indir** è l'indirizzo relativo al segmento della locazione in cui leggere il **Valore**.

Un programma per leggere la prima locazione video può il seguente:

```
Caratt = peekb (0xB800, 0);
Colore = peekb (0xB800, 1);
```

Leggere in Pascal

Come abbiamo detto prima, il Pascal non possiede un'istruzione specifica, ma un **array** che, come tale, può essere scritto e letto semplicemente indicandolo come secondo operando. L'operazione da effettuare per leggere un byte di memoria è la seguente:

```
Valore := Mem [Segm: Indirizzo];
```

...in cui **Segm** è l'indirizzo iniziale del segmento, **Indirizzo** è quello relativo al segmento della locazione in cui leggere il **Valore** contenuto nella locazione letta.

Il programma di prima diventa:

```
Caratt := Mem [47104: 0];
Colore := Mem [47104: 1];
```

Dove leggere e scrivere

Il video del PC visualizza, normalmente, 80 colonne x 25 righe. Abbiamo visto che i caratteri visualizzati, e relativi colori, sono posti nei primi 4000 bytes del segmento &HB800.

Come facciamo a calcolare l'indirizzo relativo dell'area di memoria video?

Dal momento che i caratteri ed i colori sono memorizzati alternativamente, le locazioni **pari** conterranno i codici dei caratteri visualizzati; quelle **dispari** i colori relativi.

Chiamiamo **Riga** la riga nella quale visualizzare il carattere, (numerabile tra 0 a 24) e **Col** la colonna (da 0 a 79).

Il calcolo per determinare la locazione di memoria video corrispondente è...

```
IndVid =(Riga * 160) + (Col * 2)
```

...con cui si trova l'indirizzo relativo al carattere. Va ovviamente sommato 1 per trovare anche l'indirizzo del colore relativo. Ad esempio, l'indirizzo di memoria video corrispondente al carattere nella posizione **10** (riga), **57** (colonna) di schermo è il seguente:

```
IndCar = (10*160) +(57*2)= 1704
IndCol = IndCar + 1 = 1705
```

Va ovviamente tenuto conto del segmento video, che è sempre &hB800 (oppure 47104 in decimale); gli indirizzi saranno quindi **47104:1704** e **47104:1705**.

In Basic le istruzioni corrispondenti per leggere e scrivere la locazione saranno...

```
DEF SEG = &hB800
POKE 1704, Car
POKE 1705, Col
DEF SEG

DEF SEG = &hB800
Car = PEEK (1704)
Col = PEEK (1705)
DEF SEG
```

In C:

```
pokeb (0xB800, 1704, Car);
pokeb (0xB800, 1705, Col);

Car = peekb (0xB800, 1704);
Col = peekb (0xB800, 1705);
```

In Pascal:

```
Mem [47104: 1704] :=Car;
Mem [47014: 1705] :=Col;

Car :=Mem [47014: 1704];
Col :=Mem [47014: 1705];
```

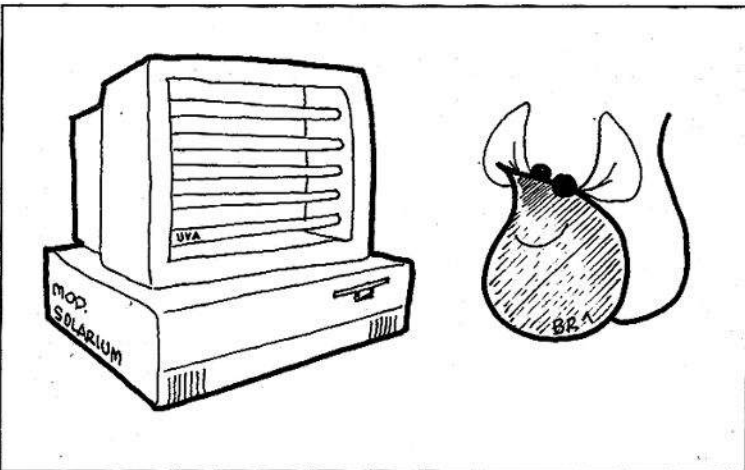
Una pratica applicazione

L'applicazione pratica di quanto studiato potrebbe essere la copia su carta della schermata video (quella che viene normalmente effettuata premendo il tasto **PrintScreen**, per intenderci).

Siccome, però, non tutti possiedono una stampante (e chi la possiede dispone anche del tasto PrintScreen...) vedremo come sia possibile registrare una schermata su disco e, in seguito, ricaricarla.

Saranno sufficienti un paio di cicli For per leggere i bytes della memoria video, e scaricarli su disco, e viceversa.

Semplici programmi che svolgono le funzioni descritte potrebbero essere quelli pubblicati in queste pagine.

In Basic esiste anche un'altra istruzione (**Screen**) che consente di leggere la memoria video senza ricorrere a Peek. L'istruzione ha il vantaggio (non richiedendo indirizzi o segmenti) di funzionare

```
20 REM POKESCRN.BAS: Programma per generare caratteri casuali sullo schermo
30 REM quindi salvare e ricaricare le schermate di testo
35 REM Versione GWBASIC e QUICKBASIC
40 REM
100 VideoSeg = &HB800        'Segmento video
105 REM
110 CLS : PRINT "  MENU": PRINT
120 PRINT "  0 - Fine programma"
130 PRINT "  1 - Generazione caratteri"
140 PRINT "  2 - Caricamento schermata 'SCREEN.SCR'"
150 PRINT : INPUT "  Scelta :"; Scelta
160 REM
170 IF Scelta <> 1 THEN GOTO 400
180 RANDOMIZE (-TIMER)   'Inizializza il generatore di numeri casuali
190 PRINT : PRINT "  Alla fine della generazione della schermata digitare":
200 PRINT "  0 - Torna al menu"
210 PRINT "  1 - Salva la schermata con il nome 'SCREEN.SCR'"

220 REM
230 PRINT : INPUT "  Quanti caratteri devo generare : "; Numero: CLS
240 FOR k = 1 TO Numero
250    Char = INT(RND(1) * 223) + 32
260    Addr = INT(RND(1) * 4000)
270    Colore = INT(RND(1) * 15) + 1
280    IF INT(Addr / 2) <> Addr / 2 THEN Addr = Addr - 1
290    DEF SEG = VideoSeg: REM Sceglie il segmento video
300    POKE Addr, Char: POKE Addr + 1, Colore
310    DEF SEG : REM Ritorna al segmento originale
320 NEXT k
330 A$ = INKEY$: IF A$ <> "0" AND A$ <> "1" THEN GOTO 330
340 IF A$ = "1" THEN GOSUB 2000: REM Salva la schermata video
350 GOTO 110
360 REM
400 IF Scelta = 0 THEN GOTO 440
405 IF Scelta <> 2 THEN GOTO 110
410 GOSUB 1000: REM Carica la schermata video
420 A$ = INKEY$: IF A$ = "" THEN GOTO 420
430 GOTO 110
440 END
500 REM
1000 REM Carica la schermata video
1005 DEF SEG = VideoSeg
1010 BLOAD "screen.scr", 0
1020 DEF SEG
1030 RETURN
1500 REM
2000 REM Salva la schermata video
2010 DEF SEG = VideoSeg
2020 BSAVE "Screen.scr", 0, 4000
2030 DEF SEG
2040 RETURN
```

con tutte le schede video. La sua sintassi è la seguente:

```
Caratt = SCREEN (Riga, Col, 0)
Col = SCREEN (Rig, Col, 1)
```

...in cui **Rig** e **Col** sono le coordinate video dalle quali leggere carattere e colore.

Rig va da 1 a 25 e Col da 1 a 80. In **Gw - Basic**, Rig può essere individuata solo da valori compresi tra 1 e 24, perché la riga 25 è riservata al Basic e non è possibile leggerla tramite Screen. In questo caso bisognerà ricorrere a Peek. In **QuicBasic** è invece possibile leggere tutte le 25 righe.

Il listato **SaveScreen**, modificato per l'istruzione Screen, potrebbe essere come quello pubblicato a parte.

Ma le risorse del Basic non sono ancora finite: possiede infatti due istruzioni che, da sole, consentono di salvare su disco e ricaricare zone di memoria desiderate: si tratta di **Bload** (Binary Load) e **Bsave** (Binary Save). Vediamo come si usano:

```
BLOAD "NomeFile", Indirizzo
BSAVE "NomeFile", Indir, Nbytes
```

...in cui **Indirizzo** è quello dal quale caricare (Bload) o salvare (Bsave) la memoria; **Nbytes**, presente solo nel caso di Bsave, è il numero di bytes da salvare. Ovviamente in tutti e due i casi andrà definito il segmento tramite Def Seg.

I due programmi di prima (LoadScrn e SaveScrn) si riducono quindi all'osso, come è possibile notare negli appositi riquadri.

Lavorando in C

La "traduzione" dei programmi visti prima per il linguaggio C è riportata a parte.

Anche in C si possono apportare migliorie. Sono infatti disponibili due istruzioni che consentono di leggere e scrivere il contenuto del video senza bisogno di specificare indirizzi e segmenti. Le due istruzioni sono **Gettext** e **Puttext**. Vediamole in dettaglio:

```
gettext (left, top, right, bottom, Destinazione);
```

L'istruzione legge la parte di schermo specificata dalle coordinate e la memorizza nell'array **Destinazione**.

Left, top sono le coordinate dell'angolo in alto a sinistra della parte di schermo da salvare. L'angolo superiore sinistro dell'intero schermo è (1, 1).

Right, Bottom sono le coordinate dell'angolo in basso a destra. L'angolo inferiore destro dell'intero schermo è (80, 25).

Destinazione è il nome dell'array di caratteri, nel quale salvare lo schermo, array che deve essere dichiarato nel programma C ed essere lungo almeno quanti sono i caratteri da salvare * 2 (dal

```
/* POKESCRN.C: Programma per generare caratteri
casuali sullo schermo, salvare e */
/* caricare le schermate di testo */
/* Versione Turbo-C */

#include <stdio.h> /* Libreria di sistema */
#include <conio.h>
#include <io.h>
#include <fcntl.h>
#include <time.h>
#include <stdlib.h>
#include <dos.h>
#include <alloc.h>

/* Variabili */

unsigned int VideoSeg; /* Segmento video  */
int Scelta;   /* Scelta da menu  */
int k;        /* Cicli FOR       */
int Numero;   /* N. caratteri da visualizzare */
int Addr;     /* Indirizzo video */
int Carat, Colore; /* Car. e col. da visual. */
char          a; /* Tasto premuto */

/* Prototipi di funzioni */
void LoadScreen(void);
void SaveScreen(void);

/* LoadScreen: Carica scher. video da disco */
/*    Segm = Segmento video
*/
void LoadScreen ()
{
  char          Schermo[4096];
  int           PicFile;
  int           k;
  clrscr();
  PicFile = open("SCREEN.SCR",O_BINARY);
  read(PicFile,Schermo,4007);
  puttext(1,1,80,25,Schermo+7);
  close (PicFile);
}; /* LoadScreen */
/* SaveScreen: Salva schermata video su disco*/
/*    Segm = Segmento video
*/

void SaveScreen ()
{
  unsigned char Schermo[4020];
  int           PicFile;

  gettext(1,1,80,25,Schermo+7);
  *(Schermo+0)=253; *(Schermo+1)=0; *(Scher-
mo+2)=184;
  *(Schermo+3)=0; *(Schermo+4)=0; *(Scher-
mo+5)=160; *(Schermo+6)=15;
  PicFile=_creat("SCREEN.SCR",FA_ARCH);
  write(PicFile,Schermo,4008);
  close(PicFile);
} /* SaveScreen */


/* Main program */

void main()
{
  VideoSeg = 0xB800; /* Segmento video */
do
{
  clrscr(); printf("  MENU\n\n");
  printf("  0 - Fine programma\n");
  printf("  1 - Generazione caratteri\n");
  printf("  2 - Caricamento schermata 'SCRE-
EN.SCR'\n");
  printf("\n  Scelta : "); scanf("%d",&Scelta);

  if (Scelta == 1)
  {
    randomize(); /* Inizial. il gen. nume. cas.*/
    printf("\n  Alla fine della generazione del-
la schermata digitare\n");
    printf("  0 - Torna al menu\n");
    printf("  1 - Salva la schermata con il nome
'SCREEN.SCR'\n");
    printf("\n  Quanti caratteri devo generare :
");
    scanf("%d",&Numero); clrscr();
    for (k=1; k<=Numero; k++)
    {
      Carat = random(223)+32;
      Addr = random(4000);
      Colore = random(15)+1;
      if ((Addr % 2) != 0) Addr++;
      pokeb(VideoSeg,Addr,Carat);pokeb(VideoSeg,
Addr+1,Colore);
    }


    do
    {
      a = getch();
    }
    while (a!='0' && a!='1');
    if (a == '1') SaveScreen();
  }

  if (Scelta == 2)
  {
    LoadScreen();
    a = getch();
  }
}
while (Scelta!=0);
}
```

```
'PCRNQB.BAS: Programma per generare
' caratteri casuali sullo schermo
' quindi salvare e ricaricare le schermate
' Versione QuickBasic
'
DEFINT A-Z
DECLARE SUB LoadScreen (VideoSeg%)
DECLARE SUB SaveScreen (VideoSeg%)

VideoSeg = &HB800        'Segmento video

DO
  CLS : PRINT "  MENU": PRINT
  PRINT "  0 - Fine programma"
  PRINT "  1 - Generazione caratteri"
  PRINT "  2 - Caricamento schermata 'SCRE-
EN.SCR'"
  PRINT : INPUT "  Scelta :"; Scelta

  IF Scelta = 1 THEN
    RANDOMIZE (-TIMER)   'Inizializza il gener-
atore di numeri casuali
    PRINT : PRINT "  Alla fine della generazione
della schermata digitare":
    PRINT "  0 - Torna al menu"
    PRINT "  1 - Salva la schermata con il nome
'SCREEN.SCR'"


    PRINT : INPUT "  Quanti caratteri devo gene-
rare : "; Numero: CLS
    FOR k = 1 TO Numero
      Char = INT(RND(1) * 223) + 32
      Addr = INT(RND(1) * 4000)
      Colore = INT(RND(1) * 15) + 1
      IF INT(Addr / 2) <> Addr / 2 THEN Addr =
Addr - 1
      DEF SEG = VideoSeg   'Sceglie il segmento
video
      POKE Addr, Char: POKE Addr + 1, Colore
      DEF SEG              'Ritorna al segmento
originale
    NEXT k
    DO
      A$ = INKEY$
    LOOP UNTIL A$ = "0" OR A$ = "1"
    IF A$ = "1" THEN CALL SaveScreen(VideoSeg)
  END IF

  IF Scelta = 2 THEN
    CALL LoadScreen(VideoSeg)
    DO
      A$ = INKEY$
    LOOP UNTIL A$ <> ""
  END IF

LOOP UNTIL Scelta = 0

END

SUB LoadScreen (VideoSeg)
  DEF SEG = VideoSeg
  BLOAD "screen.scr", 0
  DEF SEG
END SUB

SUB SaveScreen (VideoSeg)
  DEF SEG = VideoSeg
  BSAVE "Screen.scr", 0, 4000
  DEF SEG
END SUB
```

momento che vengono salvati anche i colori).

`puttext (left, top, right, bottom, Sorgente);`

L'istruzione legge l'array **Sorgente** e lo trascrive sullo schermo, alle coordinate specificate; il significato delle variabili è identico a quello per la registrazione, tranne che per Sorgente, che è il nome dell'array di caratteri destinato a contenere lo schermo. L'array deve essere dichiarato nel programma C e deve essere anch'esso lungo almeno quanti sono i caratteri da ripristinare sullo schermo * 2.

Negli array Sorgente e Destinazione il formato di salvataggio dello schermo è lo stesso di quello della memoria video, quindi i caratteri si alternano ai colori.

I programmi corrispondenti sono pubblicati a parte.

Per completare la descrizione dei programmi in C bisogna descrivere alcune istruzioni.

`Handle = open (NomeFile, Mode);`

Apre il file **NomeFile** e ne assegna il numero alla variabile intera **Handle**. Se c'è un errore, Handle varrà **-1**. Mode specifica il modo di apertura del file:

O_binary specifica un file binario
O_rdonly un file in sola lettura
O_wronly un file in sola scrittura
O_rdwr un file in lettura / scrittura

Il prototipo dell'istruzione **Open** è nel file "**io.h**" mentre quello di "**Mode**" in "**fcntl.h**".

`read (Handle, Buffer, Nbytes);`

Legge il numero di bytes specificato da **Nbytes** dal file numero **Handle** e li pone nel **Buffer**. Questo è, normalmente, un array di tipo char, che deve avere un numero di elementi uguale o maggiore al numero di bytes da caricare. Se c'è un errore di lettura, Read restituisce **-1**.

`write (Handle, Buffer, Nbytes);`

Scrive sul file numero **Handle** il numero di bytes specificato da **Nbytes**. Questi sono contenuti nel **Buffer**. Il buffer è normalmente un array di tipo char dimensionato al numero di bytes occorrente. Anche write restituisce **-1** se qualcosa non va per il verso giusto.

I prototipi delle istruzioni read e write sono contenuti nel file "**io.h**".

```
Handle = _creat (NomeFile, Modo);
```

Crea il file **Nomefile** su disco e lo apre. Se il file esiste già, lo cancella e lo ricrea. Il numero del file viene assegnato alla variabile **Handle**. Modo, invece, specifica gli attributi del file, e può essere:

Fa_arch file tipo archivio
Fa_rdonly file di sola lettura
Fa_hidden file nascosto
Fa_system file di sistema

Se **_creat** fallisce, restituisce **-1**. Il prototipo di "_creat" è contenuto nel file **"io.h"**, mentre **"Modo"** è contenuto nel file **"dos.h"**.

La grafica in C

In Turbo-C esiste un'istruzione che consente di determinare automaticamente il tipo di scheda grafica installata sul computer. Può essere utile per individuare il segmento video ed il modo di lettura / scrittura della memoria video (se, ad esempio, c'è una scheda monocromatica, non verrà registrato il colore dei caratteri). L'istruzione **Detectgraph**, e la sua sintassi è la seguente:

```
detectgraph (graphdriver, gra-
phmode);
```

Una volta attivata, nella variabile graphdriver viene restituito un valore che contiene il tipo di scheda presente nel computer in accordo con la tabella di queste pagine.

L'esempio riportato fa capire come si usa detectgraph.

Per finire, suggeriamo di esaminare la versione di Loadscreen e Savescreen scritta in **Pascal**.

Un esempio completo

Per completare l'articolo, viene inserito un esempio di programma nei 4 linguaggi proposti. Il programma, tramite menu, è in grado di:

1) Generare un numero di caratteri random a piacere sullo schermo, colorati in modo random.

2) Salvare la schermata testo con il nome **Screen.scr**.

3) Ricaricare e visualizzare la schermata testo Screen.scr.

Una volta digitato e fatto partire, il programma presenta un menu con 3 opzioni:

0) Fine programma
1) Genera caratteri
2) Visualizza la schermata

Premendo **1** viene richiesto il numero di caratteri che verranno generati e colorati in modo random. Alla fine sarà possibile premere due tasti:

0 : Ritorna al menu senza salvare la schermata video

1 : Torna al menu e salva la schermata.

Come funzionano

Gw - Basic QuickBasic

Non c'è molto da scrivere sul funzionamento del programma Basic. Le schermate video vengono salvate e ricaricate tramite Bsave / Bload, il cui funzionamento è stato descritto prima.

Due istruzioni meritano un po' di attenzione:

```
RANDOMIZE (value)
```

Inizializza il generatore di numeri casuali del PC. **Value** è il seme. Nel programma noterete **Timer** che, sempre variabile (è incaricato di contare il tempo) inizializzerà il generatore sempre in modo diverso.

```
RND(1)
```

Genera un numero casuale compreso tra 0 e 1. La riga **If Int (Addr...)** serve per generare numeri sempre pari, dato che, come abbiamo detto prima, l'indirizzo della memoria di schermo corrispondente al carattere è sempre pari.

C

Anche per quanto riguarda il Turbo - C la maggior parte delle istruzioni sono state descritte prima. Le due istruzioni da descrivere sono ancora quelle usate per generare i numeri casuali, e sono:

```
randomize();
```

Inizializza il generatore di numeri casuali prendendo come seme il Timer.

```
random (n);
```

Genera un numero casuale compreso tra 0 e **n**.

Il salvataggio ed il caricamento dello schermo vengono effettuati, come visto prima, facendo uso di Gettext e Puttext. Osservando le procedure di SaveScreen e LoadScreen si nota che la prima salva, prima dei dati dello schermo, una sequenza di 7 bytes, e precisamente:

```
253, 0, 184, 0, 0, 160, 15
```

I bytes servono per registrare le schermate nello stesso formato di Bsave del Basic. In questo modo le schermate registrate dal C possono essere ricaricate anche da Basic. LoadScreen, quindi, deve **escludere**, dalla lettura, i **7** bytes al momento del caricamento della schermata. Ciò si risolve con Putscreen, che inizia a scrivere sul video a partire da **Screen + 7**.

Pascal

In Pascal le istruzioni per generare i numeri casuali sono:

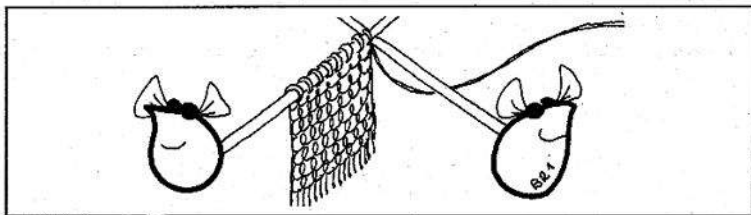
```
randomize;
```

Inizializza il generatore di numeri casuali prendendo come seme il timer; e...

```
random(n);
```

...che genera un numero casuale compreso tra 0 ed **n**. Come si vede, le istruzioni in Pascal e C sono praticamente identiche.

Le procedure LoadScreen e SaveScreen sono un miglioramento di quelle viste prima, dato che invece di salvare e leggere un byte alla volta, salvano e leggono l'intero schermo, allo scopo di velocizzare le operazioni di I/O su disco, altrimenti sarebbero troppo lente.

Anche in questo caso è stata registrata la sequenza dei 7 bytes vista prima per mantenere la compatibilità con le Bload/Bsave del Basic.

di Giancarlo Mariani

LE ISTRUZIONI DEC e JNZ

Due istruzioni Assembler faranno... fischiare il vostro computer Ms - Dos compatibile!

Finora abbiamo conosciuto (vedi C.C.C. n. **81**, **82**) alcune istruzioni dell'Assembler **80X86**, che consentono di leggere e scrivere sia i registri che la memoria (**MOV**), di sommare due numeri (**ADD**), di richiamare subroutines (**CALL**) e di saltare incondizionatamente a punti precisi del programma (**JMP**).

Tutte queste istruzioni consentono di scrivere un programma che, però, **NON** può prendere decisioni, ossia, non può prendere strade diverse a seconda di condizioni prefissate. In **Basic**, l'istruzione che permette di ottenere ciò si chiama If; ed in **Assembler**?

```
Bit  :  15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
Flag :  -- -- -- -- OF DF IF TF SF ZF -- AF -- PF -- CF

I 9 flags sono:

OF = Flag di overflow
DF = Flag di direzione
IF = Flag per abilitazione interrupt
TF = Flag per esecuzione passo-passo
SF = Flag per il segno
ZF = Flag di zero
AF = Riporto ausiliario
PF = Flag di parità
CF = Riporto
```

Innanzitutto, bisogna dire che in Assembler non è possibile, come nei linguaggi evoluti, controllare condizioni tipo *"Se la stringa A$ è uguale a PIPPO"* o cose del genere. E' invece possibile solo effettuare confronti numerici, del tipo *"Se il registro vale zero..."*, oppure *"Se la locazione XXX è maggiore di 30"*, e così via.

Le uniche istruzioni che permettono di modificare il flusso del programma che viene eseguito in un particolare momento, sono istruzioni di **salto condizionato**.

Queste istruzioni funzionano come Jmp, ma eseguono il salto solo al verificarsi di una particolare condizione del registro di stato (o **flags**) del processore. Come ricorderete (ne abbiamo parlato nei precedenti fascicoli) l'80X86 (ma anche gli altri processori) possiede al suo interno uno speciale registro, detto **Status Word** (Parola di stato) oppure Flags. Questo registro è formato da **16 bit** che, come tali, possono valere **0** oppure **1.** Di essi, **9** servono per sapere in quali condizioni si trova il processore in quel momento, mentre gli altri 7 non sono usati.

Il registro di stato, come la funzione dei flags, sono riportati nell'apposito riquadro.

Dei 9 flags, oggi ci occuperemo di quello corrispondente al **bit 6** della parola di stato, ossia del Flag di Zero (**Zero Flag**); gli altri verranno trattati successivamente.

Il flag di zero consente al programma di sapere se il risultato dell'ultima opera-

zione **matematica** o **logica** effettuata dal processore ha ottenuto come risultato zero.

Per fare un esempio, riferiamoci ad un banale programma, come quello presentato sul n. 81, che permette di effettuare la somma tra due numeri tramite l'istruzione ADD. Se il risultato di questa operazione è diverso da zero (caso più comune), il flag **ZF** è posto a zero dal processore. Se invece il risultato dell'operazione è uguale a zero (ad esempio, se si somma zero con zero), **ZF** è posto dal processore al valore unitario (1).

E' importante notare che la maggior parte dei bit della parola di stato (**CF, PF, AF, ZF, SF, OF**) sono settati o resettati automaticamente dal processore, mentre altri (**DF, IF, TF** ed ancora **CF**) possono essere settati o resettati da istruzioni Assembler.

Ritorniamo ora al nostro ZF. Le istruzioni che modificano il flag sono quelle di somma, sottrazione, operazioni tipo **AND / OR** sui bit, confronti, incrementi, decrementi, ma **NON** istruzioni tipo MOV; ciò significa che se, ad esempio, viene eseguito un **MOV AL, 0**, tale elaborazione NON influenzerà il flag ZF, qualunque sia il valore del byte interessato allo spostamento (che, nel caso specifico, vale zero: Mov Al, **0**).

Il flag ZF può essere poi controllato dal programma Assembler allo scopo di prendere determinate strade a seconda del suo valore.

Come detto prima, le uniche istruzioni che permettono di effettuare scelte sono quelle di salto condizionato. In queste pagine ci occuperemo delle due istruzioni che permettono di saltare a seconda del valore del flag ZF, ossia **JZ** (**J**mp on **Z**ero, ossia: salta se il risultato è zero, cioè se ZF = 1) e **JNZ** (**J**mp on **N**ot **Z**ero; salta se il risultato è diverso da zero: ZF = 0).

Nel riquadro a parte è descritta un'istruzione tramite cui è possibile influenzare il flag ZF, e che useremo in seguito per costruire due semplici programmi di esempio. L'istruzione è DEC; vediamo a che serve.

Alcuni esempi di istruzione JMP:

```
JZ  Label ; Salta a Label se ZF = 1
JNZ Pippo ; Salta a Pippo se ZF = 0
```

Attenzione a:

La parte di programma alla quale si salta tramite le istruzioni JZ / JNZ deve contenere corrette istruzioni Assembler 80X86, perché se si salta, per errore di calcolo, ad una parte di memoria non contenente un programma (ma, ad esempio, un messaggio alfanumerico) il computer si può inchiodare. Le due istruzioni non influenzano alcun Flag.

J(N)Z

JZ / JNZ sono i codici mnemonici delle due istruzioni, e sono, come visto prima, l'abbreviazione di Jmp on Zero e JMP on Not Zero.

Destinazione. E' l'indirizzo, o meglio, l'offset da aggiungere all'attuale contatore di programma per effettuare il salto. Questo è un salto relativo, ossia può andare solo da -128 a +127 bytes dal punto di partenza. Generalmente l'indirizzo di destinazione nelle istruzioni di salto relativo si specifica con un'etichetta.

Tutti gli indirizzi specificati nelle istruzioni JZ / JNZ fanno ovviamente riferimento al code segment CS.

JZ Destinazione (Indirizzo tipo SHORT)

JNZ Destinazione (Indirizzo tipo SHORT)

JZ salta se il risultato dell'ultima operazione matematica o logica effettuata dal processore è uguale a zero, ossia se il flag è uguale a 1.

JNZ esegue l'esatto contrario, ossia, salta se ZF è uguale a zero, quindi, se il risultato dell'ultima operazione è diverso da zero.

L'istruzione Dec

Per scrivere l'indispensabile programmino di esempio utilizzeremo il flag ZF per controllare se un determinato contatore è arrivato a zero oppure no. Un'istruzione del tipo...

```
DEC Contatore
```

...produrrà due risultati diversi a seconda del valore che possiede **Contatore**. Se Contatore valeva 1 prima di arrivare a DEC, l'istruzione lo decrementerà (quindi Contatore assumerà il valore di zero ed il flag ZF sarà settato a 1).

Se Contatore aveva un valore diverso da 1, il decremento produrrà sempre un risultato positivo, quindi il flag ZF sarà sempre resettato a 0.

Alcuni esempi di istruzione DEC

Tenendo conto che **Seg** specifica il segmento contenente i dati da trattare (e può essere **SS, DS, ES, CS**) vi suggeriamo di esaminare con cura le seguenti istruzioni.

```
DEC  CX            ; Registro a 16 bit
DEC  SI            ; Registro a 16 bit
DEC  DL            ; Registro ad 8 bit
DEC  5345h         ; Locazione assoluta di memoria
DEC  Contatore     ; Etichetta
DEC  Tabella[SI]   ; Tabella in memoria puntata da SI
```

DEC

Nel caso di trattamento di dati da un altro segmento, bisogna specificarlo prima del parametro destinazione, con il nome del registro di segmento desiderato, seguito dal carattere di **doppio punto** (:), come, ad esempio...
DEC Seg: Destinazione

DEC è il codice mnemonico (Assembler) dell'istruzione, ed è l'abbreviazione dell'inglese "DECrement", ossia decrementa. Come facilmente intuibile, l'istruzione serve per decrementare (sottrarre di un'unità) il parametro "Destinazione".

Destinazione. E' il parametro che va decrementato, ossia al quale va sottratto 1.

Destinazione può essere specificato da:
- Un qualsiasi registro del processore (AX, BX, CX....) anche nella sua forma ad 8 bit (AL, BL, CL....) tranne i registri speciali (Flags e IP, instruction pointer).
- Una locazione di memoria, anche specificata da una label.
- Una tabella in memoria indirizzata dal registro SI.

I dati trattati dall'istruzione fanno sempre riferimento al Data Segment DS.

Un semplice esempio

I due programmi proposti permettono di generare, dall'altoparlante del PC, un numero di suoni (semplici **Beep**) variabile, con un ritardo, tra un beep e l'altro, variabile anch'esso. Il Beep può essere generato, come è noto, facendo "scrivere" su video il carattere Ascii 7, anche in Basic: **Print CHR$(7)**.

La "scrittura" è ottenuta, qualunque sia il linguaggio adoperato, tramite una chiamata al DOS, e precisamente tramite l'**Interrupt 29h**. Questo si limita a scrivere su video il carattere contenuto in quel momento nel registro **AL**. Siccome, nel nostro caso specifico, il carattere vale **7**, invece di scriverlo su video, produrrà un Beep. Pertanto in Assembler, schematicamente...

```
MOV  AL, 7 ; Carattere 7
INT  29h   ; Chiamata al DOS
```

Il primo programma è riportato nell'apposito riquadro.

Esaminiamolo in dettaglio.

All'inizio vi sono le solite inizializzazioni dei registri di segmento **DS** ed **ES**, spiegate nei fascicoli precedenti.

La sequenza **MOV AL, 7** e **INT 29h**, come visto prima, servono per generare il beep.

A questo punto viene caricato, nel registro CX, il valore contenuto nei due bytes di memoria definiti dall'etichetta DELAY. L'istruzione **DEC CX** decrementa il contenuto del registro CX, e quindi, con **JNZ Ritardo1** si salta al decremento sino a quando CX è diventato 0.

In pratica viene costruito un ciclo che si ripeterà tante volte quanto vale il con-

Attenzione a:

In caso di decremento di registro ad 8 bit, DEC occuperà due bytes di memoria; nel caso di decremento di registro ad 8 bit, invece, ne occuperà uno solo.

Il decremento di locazioni di memoria è più lento di quello dei registri. Se si vuole decrementare una locazione molte volte, è meglio caricarla prima in un registro e poi decrementare il registro.

L'istruzione DEC influenza i flags AF, OF, PF, SF, ZF.

```
; BEEP1.ASM : Genera un numero variabile di BEEP
;             Genera ritardo tra un BEEP e l'altro con due
;             cicli, uno dopo l'altro
;
; Numero di BEEP = locazione NBEEP
; Ritardo        = locazione DELAY
;
cseg        SEGMENT PARA PUBLIC 'CODE'
            org 100h
            ASSUME cs:cseg, ds:cseg, ss:cseg, es:cseg

Start:
            mov  AX,CS
            mov  DS,AX            ;DS = CS
            mov  ES,AX            ;ES = CS

AltroBeep:
            mov  AL,7             ;Scrive car. 7 (Genera BEEP)
            int  29h

            mov  cx,DELAY         ;Registro CX = ritardo
Ritardo1:
            dec  cx               ;cx=cx-1
            jnz  Ritardo1         ;Se cx<>0, salta a "Ritardo1"

            mov  cx,DELAY         ;Altro ciclo di ritardo,
                                  ;uguale al precedente
Ritardo2:
            dec  cx
            jnz  Ritardo2

            dec  NBEEP            ;Decrementa numero BEEP
            jnz  AltroBeep        ;Se NBEEP<>0, salta
                                  ;a "AltroBeep"

            mov  AH,4Ch
            mov  AL,0
            int  21h              ;Ritorno al DOS

; Dati utilizzati dal programma
            NBEEP    DB 3         ;Numero di BEEP (tra 1 e 255)
            DELAY    DW 60000     ;Ritardo (tra 1 e 65535)

cseg        ENDS
            END  Start
```

tenuto delle locazioni di memoria specificate da Delay.

Una "traduzione" In Basic potrebbe essere...

For Cx = 1 To Delay : Next Cx

Il ciclo serve per generare ritardo tra un beep e l'altro. L'Assembler dell'80X86 è, infatti, talmente veloce che, se non si inserisse un adeguato ritardo, tutti i beep generati sarebbero uditi come un unico sibilo.

Successivamente è stato inserito un altro ciclo di ritardo, identico al primo, per la ragione appena vista.

Terminati i cicli di ritardo, la locazione Nbeep viene decrementata (Dec Nbeep). In seguito, se non è arrivata a zero, **Jnz Altrobeep** salta a produrre un altro beep, ed il ciclo ricomincia.

Nella locazione Nbeep deve essere inserito il numero di beep che deve produrre il programma, mentre in Delay va indicato il ritardo tra un beep e l'altro.

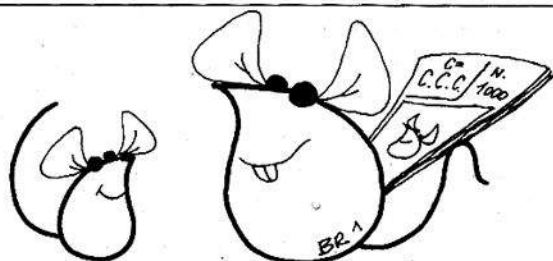
E' da notare che Nbeep è definita come **DB**, quindi è lunga 1 byte e potrà essere generato un numero variabile di Beep da 1 a 255. Delay, invece, è rappresentato da due bytes (DW = Define Word), ed il ritardo potrà variare tra 1 e 65535.

Dopo aver digitato il programma, lo si può registrare con il nome **Beep1.asm** e lo si può compilare tramite le note istruzioni:

```
MASM BEEP1;
LINK BEEP1;
EXE2BIN BEEP1.EXE BEEP1.COM
DEL BEEP1.EXE
```

Questa volta, per controllare la corretta esecuzione del programma, non avremo bisogno del debugger CV, dato che un beep è direttamente udibile dall'altoparlante del PC; potremo quindi renderci esattamente conto del corretto funzionamento del programma. Per mandarlo in esecuzione bisogna digitare, da Dos, **Beep1** e premere il tasto Return. Se tutto è in ordine, sentiremo i nostri "sospirati" beep dall'altoparlante del PC.

Provate ad eseguire il programma con diversi valori di Delay e Nbeep ricordando che, ovviamente, ad ogni cambio di valore va ricompilato il programma.

Introduciamo, ora, una miglioria. Come descritto prima, il ritardo tra un beep e l'altro è controllato da due cicli. Se il programma è fatto girare su di un computer molto veloce, come ad esempio, un **80386** con 33 Mhz di clock, il ritardo

```
; BEEP2.ASM : Genera un numero variabile di BEEP
;             Genera il ritardo tra un BEEP
;             e l'altro con due cicli nidificati
;
; Numero di BEEP = locazione NBEEP
; Ritardo        = locazione DELAY
; DELCOST = costante per il ciclo di ritardo interno
;
cseg         SEGMENT PARA PUBLIC 'CODE'
             org 100h
             ASSUME cs:cseg, ds:cseg, ss:cseg, es:cseg

Start:
             mov  AX,CS
             mov  DS,AX            ;DS = CS
             mov  ES,AX            ;ES = CS

AltroBeep:
             mov  AL,7             ;Scrive il carattere 7
                                    ;(Genera il BEEP)
             int  29h

             mov  bl,DELAY         ;Numero Cicli di ritardo
Ritardo1:
             mov  cx,DELCOST       ;Costante di ritardo
Ritardo2:
             dec  cx               ;cx=cx-1
             jnz  Ritardo2         ;Se cx<>0, salta a "Ritardo2"

             dec  bl               ;Decr. numero cicli ritardo
             jnz  Ritardo1         ;Se bl<>0, salta a "Ritardo1"

             dec  NBEEP            ;Decrementa numero BEEP
             jnz  AltroBeep        ;Se NBEEP<>0, goto "AltroBeep"

             mov  AH,4Ch
             mov  AL,0
             int  21h              ;Ritorno al DOS

; Dati utilizzati dal programma
             NBEEP    DB 3         ;Numero di BEEP (tra 1 e 255)
             DELAY    DB 15        ;N. cicli di rit. (da 1 e 255)
             DELCOST  DW 30000     ;Costante di ritardo
                                    ;per il ciclo interno

cseg         ENDS
             END  Start
```

generato dai due cicli, anche con Delay = 65535, potrebbe non essere sufficiente a separare il suono dei beep prodotti.

In questo caso si possono usare sempre due cicli di ritardo, ma invece di essere posti l'uno **dopo** l'altro, dovrebbero essere **nidificati**, ossia, l'uno **dentro** l'altro. Per fare un esempio, analizziamo le righe Basic, di immediata comprensione:

```
For Bl = 1 To Delay
For Cx = 1 To Delcost
Next Cx
Next Bl
```

Il ciclo "interno" (For Cx...) viene eseguito tante volte quante specificate dal valore Delay, che pilota il ciclo esterno (For Bl...). Da questo si deduce che possiamo far eseguire il ciclo di ritardo tantissime volte, fino ad ottenere un'adeguata separazione dei suoni. Il programma Assembler **Beep1**, modificato in tal senso, appare nel riquadro specifico.

In esso vengono usati due cicli nidificati. Il primo utilizza, come contatore di ciclo, il registro **BL**, che indicherà il numero di volte per cui deve essere ripetuto il ciclo più interno (quello comandato dal registro **CX**).

La locazione Nbeep, come nel programma precedente, contiene il numero di beep da generare. Delay contiene il numero di volte per cui deve essere ripetuto il ciclo di ritardo, mentre Delcost è il valore per il ciclo di ritardo interno.

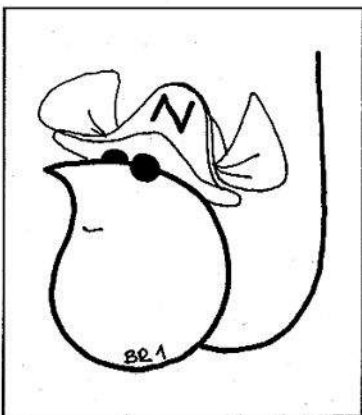
Ponendo a 255 il valore di Delay e a 65535 il valore di Delcost, è possibile ottenere un loop di 65535 * 255 = circa **17 milioni di volte**, quindi un ritardo (anche troppo) lungo per qualsiasi tipo di PC attualmente prodotto (un domani, chissà...).

Per compilare il programma dovranno essere digitate, da Dos, le stesse istruzioni di prima, con l'unica differenza relativa al nome del programma.

```
MASM BEEP2;
LINK BEEP2;
EXE2BIN BEEP2.EXE BEEP2.COM
DEL BEEP2.EXE
```

Anche in questo caso, se decidete di cambiare i valori di Nbeep, Delay e Delcost, sarà necessario ricompilare e rieseguire il programma per vederne il risultato. Bisogna fare attenzione che valori eccessivamente alti di Delay e Delcost producono un ritardo talmente lungo, soprattutto su computer lenti (basati su microprocessori 8088) da far credere che il computer si sia inchiodato.

POSTAMIGA

(a cura di Domenico Pavone)

Aspettando due tasti

Sono un accanito programmatore di Basic (Amiga), assillato da un atroce problema: come si fa a controllare con un filtro If...Then preceduto da Inkey$, se sono premuti due tasti nello stesso tempo? E inoltre: quando si preme un tasto, il computer lo scrive, aspetta un po' e poi lo stampa ripetutamente. Domanda: Come faccio ad eliminare il primo lasso di tempo?

(Andrea Cattò - Milano)

Una totale contemporaneità nella pressione di due tasti, quantomeno in basic stretto, non sarebbe teoricamente implementabile con **Inkey$**. Tuttavia si può simulare con sufficiente affidabilità qualcosa del genere "giocando" un po' con gli operatori logici associabili alla condizione If. Vediamolo in pratica, risulterà subito chiaro. Si consideri breve listato di questa stessa pagina.

Come si noterà, viene applicata due volte la funzione Inkey$, per recepire ogni volta un singolo carattere dalla tastiera. Pur se la contemporaneità non è assoluta, mandando in esecuzione il programma si noterà come di fatto, premendo (in questo esempio) i tasti **A** ed **X** nello stesso momento, si otterrà la conferma a video dell'avvenimento. Premendoli, invece, in momenti diversi, il programma continuerà a restare in attesa degli eventi.

In pratica, dopo il primo Inkey$, la condizione If...Then accetta indiscriminatamente la pressione di A oppure X per passare al secondo Inkey$. Qui, viene di nuovo vagliata la corrispondenza con uno dei due tasti, dopodiché inizia il vero e proprio "filtro". Costituito, si noti, da una condizione **OR** (vera se uno dei due elementi è vero), che vaglia però due termini racchiusi tra parentesi, ognuno dei quali opera un raffronto **AND** (vero solo se entrambi i termini sono veri).

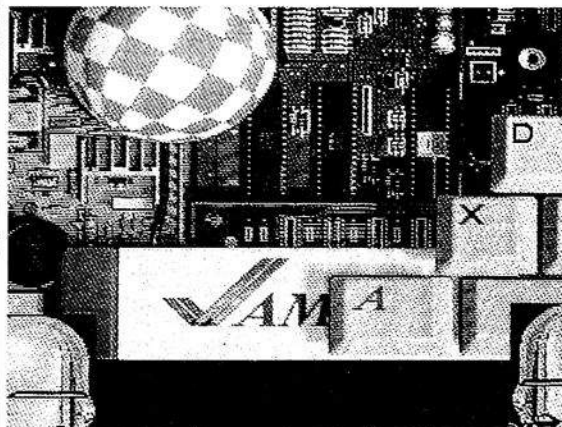
```
IPRINT "Premi A ed X contemporaneamente"
loop:
a$=UCASE$(INKEY$)
IF a$ <> "A" AND a$ <> "X" THEN loop

b$ = UCASE$(INKEY$)
IF b$ <> "A" AND b$ <> "X" THEN loop

IF (a$="A" AND b$="X") OR(a$="X" AND b$="A") THEN
 PRINT "Hai premuto A ed X assieme!"
 PRINT "Vinci una bambolina di peluche."
END IF
```

Quest'ultima esigenza è dovuta al fatto che non è possibile stabilire con precisione se il tasto A è stato premuto effettivamente prima o dopo di X, o almeno non serve saperlo: l'iportante, come richiesto dalla lettera, è che entrambi lo siano stati. Per stabilirlo, comunque, si provi ad aggiungere un banale **Print a$; b$** prima di **End If**, in fondo al listatino.

Provando più volte ad attivare il programma, si noterà come l'esito varierà tra **AX** e **XA**, a seconda... dell'estro del momento.

Non si dimentichi, prima di premere i tasti, che va eventualmente attivata la finestra di output del basic (clickandovi col mouse) se non lo è già.

Quanto all'intervallo che intercorre tra la pressione di un tasto e la sua ripetizione in automatico, può essere variato tramite il programma **Preferences**, contenuto nella directory Prefs del disco Workbench.

Dopo la sua attivazione con il consueto doppio click, è possibile agire sul "cursore" mobile contrassegnato dalla scritta **Key Repeat Delay**, che modifica, appunto, la durata dell'intervallo prima che inizi la ripetizione del tasto. Spostandolo verso sinistra (**short**) si otterrà un più rapido (fino a quasi immediato) avvio

del Repeat, verso destra (**Long**) invece un incremento dell'intervallo.

Amigo in arrivo

Sto per acquistare un Amiga, al momento ho ancora un C/128 Commodore. Vi ho scritto parecchie volte senza avere risposta, potreste almeno stavolta accontentarmi? I miei dubbi: Sento parlare di due versioni di Amiga, la 1.2 e la 1.3. Quale delle due mi dovrei procurare? Si può collegare Amiga ad un Tv color dotato di presa Scart? Esistono il Turbo Pascal ed il Logo per questo computer?

(Eugenio Tacchini - Rivergaro)

Ebbene sì, per questa volta ci siamo lasciati impietosire, e rispondiamo.

Scherzi a parte, molto spesso non si risponde a tutte le lettere che pervengono in redazione, non tanto per cattiva volontà (della quale comunque non manchiamo...), ma per l'esigenza di privilegiare quesiti che rivestano un interesse il più generale possibile, non solo per il lettore che li pone. Questa è la regola. Per le eccezioni.... passare alla cassa (eh eh eh).

Bando alle ciance. Il problema della versione del sistema operativo di Amiga, si pone solo se si pensa di acquistarne uno usato. In negozio, limitatamente ai modelli 500 e 2000, non si troverà altro che la versione **1.3** (salvo improbabili giacenze di annata), ovvero la più recente. Rivolgendosi all'usato, potrebbe anche capitare una **V 1.2**. Non è che sia proprio da gettar via, intendiamoci, ma chiaramente in questo caso andrebbe operato quantomeno un netto decurtamento del prezzo.

Amiga può connettersi tranquillamente ad un tv-color dotato di presa Scart, purché sia possibile uno switch (sul Tv) che in qualche modo attivi il segnale da essa proveniente, per esempio qualcosa come **Extern** presente, in alcuni modelli, perfino sul telecomando.

Quanto ai linguaggi: il **Logo** non presenta alcun problema, anzi ne esistono addirittura delle versioni di Pubblico Dominio. Il **Turbo Pascal**, se si intende quello strettamente **Borland**, è implementato **solo** su "macchine" Ms Dos, ambiente peraltro agibile anche su Amiga attraverso opportuni emulatori (prassi comunque sconsigliata). Tuttavia esiste il **Kick Pascal**, giunto alla sua versione **2.0**, del tutto simile al Turbo P. anche da un punto di vista... estetico, e in ogni caso validissimo per programmare in Pascal.

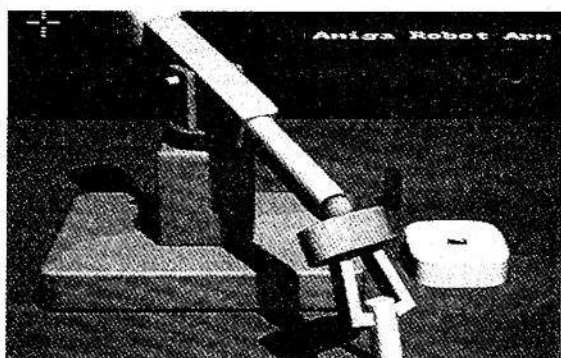
Start grafico

Sono un ragazzo di 15 anni, possessore di un Amiga 500, e non ho mai scritto per paura di non ricevere risposta ad una domanda troppo semplice. Vorrei sapere come posso inserire un file IFF in un file di avviamento. Ho cercato invano di farlo, ma forse mi manca qualcosa...

(Enrico Ferranon - Alghero)

Una domanda non è mai "troppo" semplice o elementare. Non tutti i lettori possono essere già esperti, e la rivista è fatta anche per loro. Ma veniamo al dunque.

Dando per scontato che per "file IFF" si intenda un'immagine grafica in quel formato (IFF è uno standard non necessariamente legato alle immagini!), il problema può essere affrontato con molta semplicità, soprattutto se si è alle prime armi.

Come arguisce il giovane Enrico, probabilmente manca qualcosa: un programma che visualizzi questo tipo di grafica. Prima di ogni altra cosa, è quindi necessario procurarsi una delle svariate versioni di **Show** (o **Display**), preferibilmente tra quelle che consentono anche di prestabilire una certa **durata** della visualizzazione. Reperire questo file non dovrebbe essere difficile, ed è stato anche incluso in una della nostre raccolte software, **Amigazzetta 6**.

Come ovvio, il file Show andrà copiato nel disco che si vuole dotare di autostart "visivo", preferibilmente nella directory **C**, in modo che venga rintracciato senza problemi dal sistema. Naturalmente, da qualche parte nel dischetto, deve essere presente **anche** il file IFF. Come ultimo passo, non resta che modificare la **startup-sequence**, ovvero il file-comandi (batch) contenente tutte le istruzioni che vengono eseguite nella procedura di **boot**. Tutte queste operazioni, se si è alle prime armi, è consigliabile eseguirle su una **copia** del disco **Workbench**, dal quale magari si è tolto qualche file (il Notepad, per cominciare, va benissimo) per fare un po' di spazio.

Tornando alla Startup-sequence, per modificarla *ad hoc* occorrerà eseguire le seguenti operazioni:

* Aprire una finestra Shell biclickando nella relativa icona.

* Digitare al suo interno `Cd S:` (e Return) e quindi `Ed Startup - sequence.` Chi fosse già avvezzo all'uso di un altro editor, può adoperare il suo preferito, purché salvi i

testi in formato strettamente Ascii.

* Inserire all'inizio del batch file qualcosa come **Show Pic**, ammesso che Pic sia il nome dell'immagine da visualizzare. Per essere più sicuri che tutto funzioni a dovere, è preferibile specificare, in ogni caso, il percorso del file IFF, per esempio con `Show MioDisco:Pic` se si trova nella directory principale del floppy di nome **"miodisco"**.

* Uscire da Ed premendo prima **Esc**ape e poi **X**, operazione che salverà, con lo stesso nome, il file Startup-sequence, ma con la modifica apportata.

Tutto fatto. Il disco dovrà naturalmente essere autobootabile, come appunto il Workbench; se non lo fosse, basterebbe comunque un comando `Install Miodisco:`.

Una volta lanciato il sistema con il disco modificato, apparirà l'immagine IFF prescelta, che resterà sullo schermo (a seconda della versione di Show adoperata) fino alla pressione di un tasto del mouse, o per un certo tempo prestabilito.

Come ovvio, l'istruzione Show Pic vista nell'esempio (la più "standard", per così dire) andrebbe comunque adattata alla sintassi del programma di visualizzazione adoperato, che potrebbe anche non chiamarsi Show.

C'è da aggiungere che, oltre a questa tecnica elementare, pur se efficace, esiste anche la possibilità di creare lo stesso effetto adoperando svariati programmi circolanti nell'ambito del Pubblico Dominio, che installano direttamente nel **bootblock** un minifile che provvede alla visualizzazione, talvolta anche **animata**, di un'immagine.

Una volta capito il meccanismo, comunque, non c'è che l'imbarazzo della scelta.

Cercasi libro

Ho chiesto in alcune librerie il volume "Linguaggio C" di Callegari-Feletto da voi recensito sul n. 73, ma mi sono sempre sentito rispondere che la casa editrice (GR) era sconosciuta. Forse l'avete scritta male? O il libro è esaurito?

(Fabrizio Bazzo - Gorizia)

No, non è stato commesso alcun errore di stampa. Non trattandosi di una nostra pubblicazione, possiamo solo consigliare di rivolgersi a librerie specializzate o comunque ben fornite come la **Hoepli di Milano**, oppure contattare direttamente l'editrice all'indirizzo riportato nella recensione sul n. 73.

Spazi, maledetti spazi

Mi trovo sempre in difficoltà quando devo adoperare dei nomi che contengono degli spazi per dischi, files, e soprattutto per dare il nome alle finestre shell. Possibile che si debba ammattire così? Non c'è un metodo per eliminare questo inconveniente?

(Pino Gervasi - Salerno)

L'unico metodo per evitare problemi, è quello di attenersi rigorosamente a quanto prescritto da Amiga Dos, a meno di non tagliare la testa al toro e... non adoperare gli spazi. Per inciso, non sono solo gli spazi a provocare certi inconvenienti, ma anche il simbolo *uguale* (=), *più* (+), nonché il *punto e virgola* (;), tutti da trattare esattamente come se in un nome di file, disco, eccetera, fossero presenti degli spazi.

L'argomento, in effetti, non viene molto affrontato dal manuale fornito in dotazione al computer, per cui proviamo a stilare una specie di decalogo del Dos (non proprio in dieci punti... ma l'importante è capirsi) per aggirare qualunque ostacolo.

1) Un nome di file, directory, o disco non può superare i **30 caratteri**. Gli unici simboli tassativamente non ammessi sono i due punti (:) e la barra obliqua (/).

2) Quando in un nome di file, ed eventuale suo percorso, è presente uno **spazio**, un simbolo di **eguaglianza**, di **somma** o di **punto e virgola**, allora questo deve essere racchiuso tra **doppi apici** (virgolette ").

Tradotto in pratica: supponiamo di avere un file di nome *"Roba Schifosa"* nella directory corrente, e di volerlo copiare in ram disk. Il comando andrà così impartito:

```
Copy "Roba Schifosa" to Ram:
```

E fin qui tutto semplice. Attenzione, però: se fosse necessario includere il percorso del file, questo andrebbe **TUTTO** racchiuso tra virgolette, **non solo** il nome vero e proprio. Per fare un esempio, si pensi alla stessa operazione appena vista, solo che il nostro file si trova nella directory *"avariati"* del disco posto nel secondo drive. La forma sintattica diventerebbe stavolta...

```
Copy "Df1:avariati/Roba Schifosa" to Ram:
```

Lo stesso accorgimento, come ovvio, vale per qualunque altro comando del Dos, e anche se a contenere lo spazio fosse il nome di una directory, e non il nome di un file; per esesempio...

```
Cd "Mio disco:avariati"
```

Riferendoci direttamente al problema del lettore, l'istruzione per aprire una finestra Shell con un nome composito dovrebbe quindi (p. es.) assumere una forma del genere:

```
Newshell "newcon:0/ 0/ 640/ 250/Mie Cose"
```

Si presti attenzione, però, a non esagerare con i doppi apici. Se ad un comando sono associati dei parametri legati a particolari parole chiave, queste ultime non devono essere incorporate nelle virgolette. Così, se per esempio si volesse formattare un floppy con nome *Bingo Bongo*, basterebbe racchiudere tra doppi apici solo la specifica del nome:

```
Format Drive Df1: Name "Bingo Bongo"
```

3) Tra i caratteri permessi nel definire il nome di un elemento Dos (file, disco, directory, ecc.), vi sono anche i doppi apici. In questo caso, come ovvio, occorre segnalare ad Amiga che il simbolo non deve essere interpretato a fini particolari, ma proprio come un qualunque altro carattere. Giusto per complicare l'esistenza di noi indifesi utenti, allo scopo va adoperato un **asterisco** (*) prima delle virgolette che faranno parte del nome, ed il tutto andrà poi racchiuso tra altri doppi apici. Supponiamo di voler rinominare il disco presente nel secondo drive in **"bleah!"**, virgolette comprese. Se ci limitassimo ai soli asterischi, vale a dire qualcosa come...

```
Relabel Df1: *"Bleah!*"
```

... il nuovo nome conterrebbe le virgolette, ma anche gli asterischi! La forma corretta sarebbe dunque...

```
Relabel Df1: "*"Bleah!*""
```

4) Sempre più difficile: e se si volesse inserire un asterisco in un nome contenente degli spazi? Se dopo aver letto quanto sopra non si è ancora deciso di lasciar perdere spazi, virgolette, e amenità simili, forse questo vi convincerà definitivamente.

Poniamoci come obiettivo di assegnare stavolta il nome **Bleah! * Bleah!** al disco in questione, asterisco e spazi

compresi, naturalmente. Se si adopera un normale...

```
Relabel Df1: "bleah! * bleah!"
```

... come risultato (lo si provi su un disco formattato) l'asterisco non verrebbe inserito nel nome. Per ottenere quanto desiderato, sarà necessario anteporre all'asterisco... un altro asterisco. Ovvero:

```
Relabel Df1: "bleah! ** bleah!".
```

E infine, in un pirotecnico finale dossiano, proviamo a mettere tutto assieme: spazi, asterischi, e chi più ne ha più ne metta: diamo al solito disco il nome **"Bleah! * Bleah!"**, comprese le virgolette e l'asterisco. Il comando diventerà stavolta:

```
Relabel df1: "*"bleah! ** bleah!*""
```

Una domanda è d'obbligo: ne varrà la pena?

Icone, dischi e... Kafka

Ho un Amiga 500 inespanso e con un solo drive. Ho tre domandine da porvi, due tecniche ed una un po' meno:

1) Come si può ottenere un'icona con un'immagine in "negativo" diversa dall'icona non selezionata, usando l'IconEd?

2) Alcuni giochi, se inseriti da Workbench, segnalano un Df0:Ndos. Sono dischi esaminabili da programmi tipo Diskx? Anche i comuni mortali possono fare dischi del genere?

3) Il signor Gregor Samsa, che ha recensito alcuni prodotti sulla vostra rivista, sembra uscito da un libro di Kafka: ditemi che si tratta solo di uno STRANO caso di omonimia...

(Andrea Brighenti - Trento)

Per tre domande stringate, tre altrettanto concise risposte:

1) Per avere in negativo l'immagine di un'icona selezionata (dopo il click su di essa, per intenderci), è sufficiente disegnarla con l'IconEd, dopodiché scegliere l'opzione **Inverse** dal menu **Hilite**. Così facendo, però, resterà immutato il disegno dell'icona dopo la selezione, ovvero il click su di essa. Volendo invece ottenere un'immagine diversa, non c'è altra soluzione che disegnarla ex novo, magari con i colori in "negativo" (se proprio li si vuole), e salvarla con un nome diverso dall'icona principale. Ciò fatto, e badando a mantenere delle corrette proporzioni tra le due immagini, è poi possibile utilizzare **IconMerge** (directory **Tools** del disco **Extras**) per fonderle assieme facilmente, ed ottenere l'effetto desiderato. Non ci soffermiamo più di tanto a descrivere quest'ultimo programma, in quanto il suo uso è ampiamente documentato mel manuale del Dos in dotazione al computer.

Semmai, può risultare più comodo usare altri icon-editor (consigliabile), uno tra tutti e forse il migliore **Icon Paint**, che consentono elaborazioni più sofisticate e realizzabili in un unico ambiente di lavoro.

2) I floppy segnalati come **NDos**, però regolarmente bootabili, sono in genere frutto di sofisticate manipolazioni per velocizzare (o proteggere da occhi indiscreti) particolari programmi. O ancora, potrebbe trattarsi più semplicemente di dischetti formattati adottando il **F**ast **F**ile **S**ystem (**FFS**), anche se poco produttivo non in ambito hard disk. I comuni disk-editor normalmente non possono accedere alle informazioni memorizzate in questi floppy "anomali", e una loro manipolazione è sì alla portata dei mortali, ma non certo... comuni.

3) Il signor Gregor Samsa è proprio uscito da un libro di Kafka, ed ha deciso di prestare occasionalmente i suoi servigi per la redazione di CCC. Unico problema: stanarlo ogni volta da sotto il computer di chi scrive, dove si ostina a rifugiarsi tra un orrido brulichio di zampe...

Action Replay e 2000

Ho letto qualche mese fa la vostra recensione della cartuccia Action Replay, e ne sono rimasto entusiasta. Però ho un Amiga 2000, così devo soffrire in silenzio. Possibile che non esiste niente del genere per il mio computer? Vuol dire che devo comprarmi un A 500 per adoperarla?

(Saverio Barca - Milano)

Qualcosa "del genere" per l'Amiga 2000 esiste, e si chiama... **Action Replay 2**. A distanza di un paio di mesi dalla pubblicazione di quella recensione, è infatti stata prodotta una release aggiornata della cartuccia, della quale esistono due versioni: una per A 500 ed una per A 2000. Considerata la provenienza della lettera, non dovrebbero esserci problemi sulla sua reperibilità, in quanto messa in vendita presso una delle ditte comprese tra i nostri inserzionisti (Flopperia Milano), che tra l'altro opera anche per corrispondenza.

Trasferimenti grafici

Ho realizzato molti disegni col mio Amiga, adoperando soprattutto Deluxe Paint III. Vorrei fare in modo che si possano vedere anche sui computer Ibm compatibili, ma non so esattamente come fare, anche se so trasferirli col Dos2Dos. E' vero che i Pc non leggono l'IFF?

(Silvano Borsi - Roma)

Prima di rispondere, vediamo di chiarirci un po' le idee. L'IFF, come risaputo, non è altro che uno standard che codifica un metodo di memorizzazione di dati, nel caso specifico riguardanti la grafica. In quanto tale, non sono i Pc oppure gli Amiga a "leggere" l'Iff, ma dei programmi in grado di riconoscere questo formato.

Il che significa: dato per scontato il trasferimento da un computer all'altro (col **Dos2Dos**, via cavo seriale, o anche via modem), se esiste un programma del genere per i Pc, i file Iff potranno benissimo essere visualizzati anche in quell'ambiente senza alcuna modifica.

E anzi, senza andare troppo lontano, un programma del genere esiste proprio, e si chiama guarda caso... **Deluxe Paint II**.

```
SetPatch nil:
cd c   ;per velocizzare la ricerca
resident CLI L:Shell-Seg SYSTEM pure add
resident c:Execute pure
mount newcon:
FF NIL: -0
assign t: ram:
assign clips: ram:
;-----------------------------------
ask "*ec Tastiera italiana? (Y/N)"
if warn
 sys:system/Setmap i
endif
;-----------------------------------
ask "*ecHai un orologio hardware? (Y/N)"
if warn
 setclock load
else
;-----------------------------------
 failat 100
 lab manual
 echo "*ecInserisci data e ora in forma:"
 echo "      GG-MES-AN OR:MN:SC"
 echo " "
 echo "(GG=giorno MES=mese AN=anno)"
 echo "(OR=ora MN=minuti SC=secondi)"
 echo " "
 date nil: ?
 if error
  echo "*ec Formato scorretto!"
  echo " "
  ask "Vuoi riprovare? (Y/N)"
  if warn
    skip manual back
   endif
  endif
  echo "*ec"
  date
  echo ". "
  ask "E' corretta? (Y/N)"
  if not warn
   skip manual back
  endif
  failat 10
endif

;-----------------------------------
echo "*ecDIGITA:"
echo "Y (+return) per ambiente Workbench"
echo "N (+return) per ambiente Shell"
ask ""
if warn
 loadwb -debug
 endcli
endif
;-----------------------------------
if not warn
 ask "*ecTutti i comandi in Ram? (Y/N)"
 if warn
  copy sys:c ram:c
  assign c: ram:c
 endif
 cd sys:
 Newshell Newcon:0/0/640/255/SHELL
 endcli
endif
```

Il discorso cambia, però, se si parla di diffusione dello standard. Nel senso che, allo stato attuale, quello che per Amiga è pane quotidiano (l'Iff, appunto) sugli Ms - Dos compatibili è, di fatto, meno diffuso, per cui può risultare più conveniente trasferire i propri disegni dopo averli "tradotti" in un formato più usuale nel mondo Ms-Dos: PaintBrush (**Pcx**), o anche il meno professionale (ma anch'esso di larga diffusione) **Gif**.

Per trasformare un file prodotto dal DPaint in altri formati, è necessario ricorrere ad appositi programmi, il più versatile e completo dei quali è senza ombra di dubbio **ImageLink**, che consente di tradurre praticamente da un qualsiasi formato all'altro, comprese alcune codifiche molto particolari legate ad Amiga, ai Pc, e addirittura anche ai **Macintosh** o ad alcuni digitalizzatori video: Iff, Gif, Pcx, Sculpt, Sun, Digiview, Caligari, Targa, Turbo Silver... e chi più ne ha più ne metta.

Per di più consente anche, in fase di adattamento, di modificare la risoluzione dell'immagine in quella che più aggrada.

Molto utile, nel caso del nostro lettore, se per esempio si intendesse trasportare in ambiente Pc un'immagina grafica in Hi Res con risoluzione verticale differente da quella (sempre a titolo di esempio) adottata dalle schede **Vga**.

In definitiva, basterà quindi tradurre l'immagine Iff prodotta su Amiga nel formato desiderato (preferibilmente Pcx), quindi trasferirla fisicamente così com'è in un floppy Pc tramite il dos2dos (o con altri mezzi), dove potrà essere visualizzata da programmi grafici che adottano quello standard, o programmi più generici sullo stile dell'amighiano **Show**.

Tutto con un batch

Sono un principiante con Amiga, ma sto cercando di aggiornarmi seguendo i vostri articoli sul Dos. Quello che vi chiedo è di pubblicare un file in modo che posso decidere dall'inizio se adoperare un disco con il Workbench oppure con la Shell, e che vada bene anche per chi, come me, ha ancora un solo drive e non è molto esperto.

(Paolo Cardella - Palermo)

Detto, fatto. Si può utilizzare un batch file come quello pubblicato a parte in queste pagine, da sostituire alla startup - sequence.

Ovvero: prima effettuare una copia del disco Workbench, quindi operare la sostituzione della startup-sequence.

Per chi è ancora alle prime armi, non adopera una qualche utility come **DiskMaster**, e per giunta non possiede un secondo drive (pressoché indispensabile!) ecco il detta-

glio delle operazioni da eseguire (i più evoluti potranno "scremare" quanto a loro non necessario):

* Effettuare una copia del Workbench, rinominandola (p. es.) in **WB**. Se anche questa operazione dovesse risultare complessa... beh, si dia un'occhiata al manuale oppure si adoperi qualche copiatore (quello l'abbiamo tutti da qualche parte, vero?...).

* Aprire una finestra Shell, ed editare (= copiare) il batch file pubblicato. Il che si traduce nell'uso di un'editor Ascii (il Notepad **non** va bene!) o, se non si è in grado di utilizzarlo, adoperando **Ed**. In questo caso occorre prima impartire `Cd Wb:s`, seguito dopo un Return da `Delete Startup-Sequence`, per eliminare il vecchio file di start. Il tutto, se si ha un solo drive, accompagnato (ahivoi) da svariati cambi di dischetto. Con due drive, basterà invece inserire Wb nel secondo drive, e tutto procederà speditamente. Dopo questi preliminari, si potrà finalmente impartire `Ed Startup-sequence`, e copiare fisicamente il nostro batch file (si veda eventualmente AmigaFacile del n. 78 per i comandi di Ed). Finita la copia, uscire da Ed con Escape X, attendere qualche secondo che il file sia effettivamente memorizzato, e adoperare il disco WB per un nuovo boot di Amiga.

Se non si sono commessi errori, durante la fase di lancio sarà ora possibile scegliere il tipo di **tastiera** adoperata, il settaggio della **data** e dell'**ora** se non si dispone di un'espansione con orologio, nonché **optare** tra ambiente Workbench e Shell. In quest'ultimo caso, per velocizzare la ricerca dei comandi, ma soprattutto se si dispone di un solo drive, si può anche decidere se avere i comandi in Ram.

Il tutto automaticamente, e solo con qualche input da tastiera limitato alla pressione di **Y** (per yes) oppure **N** (per No) sempre seguito dal Return.

Quanto al batch, questo riassume in pratica alcune applicazioni già pubblicate singolarmente sulla rivista, con piccole modifiche adatte al nuovo scopo. La prima parte si limita ad eseguire quanto normalmente fa la startup del normale disco workbench, ovvero installare il device Newcon: funzionale all'uso della Shell (si veda anche AmigaFacile n. 79), ed assegnare T: e Clips: alla Ram: per evitare continui accessi al disco quando si adoperano editor che necessitano di questi device (se ne riparlerà).

A seguire, vengono proposte le varie scelte sfruttando il comando **Ask**. Poco da dire sull'insieme delle operazioni, che si limitano ad applicare intensamente quanto più volte descritto nei vari appuntamenti con questa rubrica o con AmigaFacile.

Da notare l'uso dei caratteri "*ec" associati ad Echo oppure alla stringa stampata da Ask. Si tratta in pratica della traduzione del comando Escape + c (*e = Escape), che provoca uno svuotamento dello schermo.

Come più volte ribadito, si presti attenzione al formato della data da inserire se non si dispone di un orologio hardware: il mese va precisato con le prime tre iniziali del termine inglese.

Per chi proprio non masticasse questa lingua, ecco un prontuario da consultare all'uopo:

JAN	Gennaio
FEB	Febbraio
MAR	Marzo
APR	Aprile
MAY	Maggio
JUN	Giugno
JUL	Luglio
AUG	Agosto
SEP	Settembre
OCT	Ottobre
NOV	Novembre
DEC	Dicembre

La fase di immissione della data è comunque fornita di un controllo di errore, affidato all'uso di **Failat** (si veda AmigaFacile sul numero scorso).

Per quanto riguarda la copia dei comandi in Ram, questa operazione potrebbe risultare apparentemente scomoda, in quanto allunga discretamente i tempi della procedura di start. Tuttavia, se si dispone di un solo drive, è l'*unica* alternativa disponibile per operare con una certa tranquillità. In questo caso, comunque, occorre anche fare i conti con la memoria disponibile, che verrebbe decurtata di circa 220 Kb.

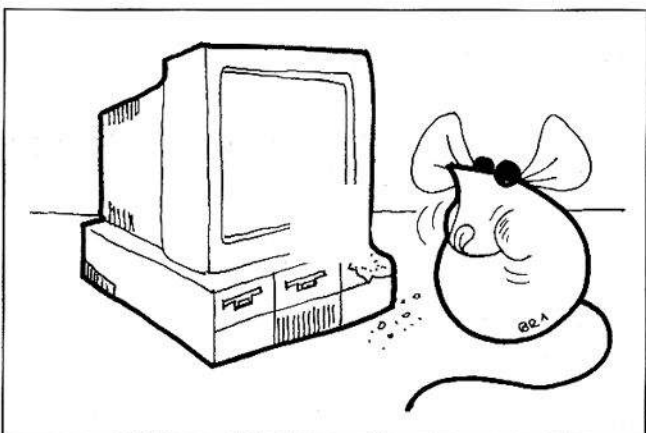
Non rimane da aggiungere che una considerazione sull'inevitabile rallentamento del boot provocato dai vari input. Chi già possiede una certa dimestichezza con il Dos (ma anche gli altri possono cimentarsi, provare non nuoce), può eliminare eventuali Input inefficaci sulla configurazione posseduta, lasciando solo quei comandi che svolgano i compiti che interessano maggiormente. A chi avesse invece problemi di memoria, si consiglia di sostituire al comando...

```
Copy sys:c ram:c
```

...che copia l'intera directory C in Ram Disk (creandovela), una serie più mirata di "copy", che trasferisca solo quei comandi di più largo uso. In questo caso, risulterà prima necessario creare una directory **C** in ram con un banale `Makedir Ram:c`, seguito da una sequela di...

```
Copy sys:c/xxxx Ram:c
```

...con xxxx da sostituire con il nome del file da copiare. Si ricorda, a questo proposito, che sono indispensabili i comandi **Run, Execute, Newhell, Endcli**.

Un consiglio per i principianti: si provi per un po' di tempo ad operare con il nuovo disco lasciando inalterata la startup - sequence qui proposta. Una volta "fatta la mano" ed assicuratisi del suo corretto funzionamento (= la si sia copiata senza errori), provare ad inserire singole modifiche, in modo da poterne valutare l'efficacia.

ENV:

Il Dos e le variabili

In tutti i linguaggi di programmazione, il ruolo delle variabili è determinante. Anzi, potremmo dire che senza di esse non si potrebbe praticamente far nulla.
AmigaDos, fino ad ora, è stato affrontato come qualcosa che si avvicina ai linguaggi di programmazione, dotato di suoi comandi (tra l'altro incrementabili) e della possibilità di organizzarli in batch files, che in pratica possono essere genericamente ricondotti alla struttura di un programma, ma senza implementare alcuna variabile interna, o "di ambiente" come spesso vengono definite.
Di fatto, però, per una gestione evoluta del dos, non si può fare a meno di curare la possibilità di implementare una qualche forma di variabile, che aggiunga ulteriore flessibilità a quanto ottenibile con gli **Script File** (sono sempre i batch, niente paura). Pur se con certi limiti, e con una gestione non proprio semplicissima per chi è agli inizi, dalla versione **1.3** del sistema l'uso di elementi variabili è stato reso operativo dalla comparsa del device logico **Env:**, dai comandi **Setenv** e **Getenv**, nonché da alcune possibili associazioni di variabili alla condizione **If**.
In queste pagine si vedrà come adoperare questa feature di AmigaDos, di solito poco sfruttata, e che in futuro verrà ulteriormente perfezionata dalla versione 2.0, sempre che arrivi sui modelli "normali" di Amiga (leggi A - 500 e A - 2000).

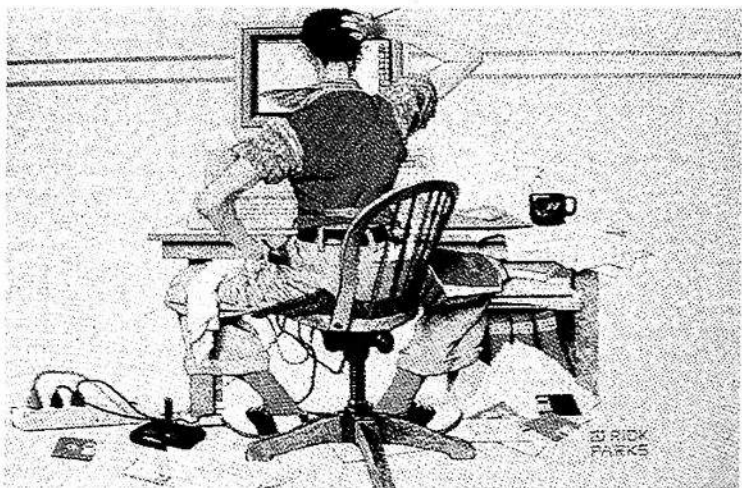
ENV:

E' il device logico predisposto alla gestione delle variabili in ambiente Dos. Questo dispositivo, per risultare attivo, deve essere assegnato a qualcosa di esistente, cosa che avviene automaticamente quando si adopera, per esempio, il disco **Workbench 1.3** per attivare Amiga.
Per trovarne traccia, è sufficiente impartire un banale **`type S:StartupII`** da una finestra Shell. Si potrà constatare come questo batch, che viene a sua volta lanciato dalla Startup-Sequence del Workbench, contiene due istruzioni...

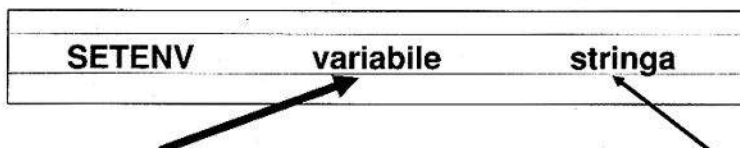
```
Makedir ram:env
Assign env: ram:env
```

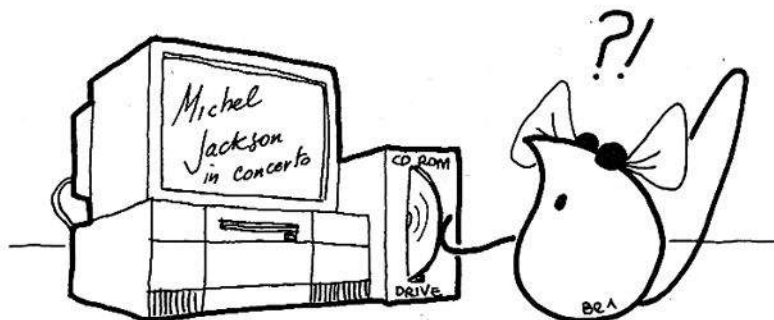
... che abilitano appunto il device, assegnandolo ad una directory di nome **Env** creata in **Ram Disk**. Per propri scopi, si può naturalmente anche modificare il nome della directory (non quello del device!), o addirittura assegnarlo genericamente alla Ram con...

```
Assign Env: Ram:
```

Nel luogo designato dall'assegnazione, andranno a finire fisicamente le "variabili", che altro non saranno che dei files ASCII il cui contenuto rappresenta la variabile vera e propria. Vediamo di chiarire meglio.
In un linguaggio come il Basic, è possibile adottare una forma tipo **`A$="pippo"`**, in modo che ogni volta che si adopera A$ venga, in realtà, adoperato il suo contenuto, ovvero "pippo". In un secondo momento, si potrà sempre cambiare il contenuto della variabile, con un ennesimo **`A$="Pluto"`**.
Ebbene, in ambiente dos la cosa procede così: intanto non esiste una differenza tra variabili numeriche, stringa, eccetera. Quando si assegna (vedremo tra breve come) "pippo" alla variabile (p. es.) `Vattelapesca`, si creerà concretamente un file Ascii di nome Vattelapesca nella directory assegnata ad Env: (in Ram:env, nel caso del workbench). Il testo contenuto in questo file sarà (guarda caso) "pippo". Una volta assegnata (si veda l'esempio di queste pagine), una variabile può essere citata nell'ambito di un batch file facendola precedere dal simbolo di dollaro (p. es. **`$pippo`**).
In definitiva, se si intendono adoperare delle variabili, occorre anzitutto assegnare a ENV: una collocazine fisica (directory o altro device che sia) una volta per tutte, meglio se nella startup-sequence, tramite il comando **Assign**. Quindi ricorrere ai comandi o alle tecniche di seguito descritte per assegnare un contenuto alle variabili. La loro memorizzazione, come pure il richiamo di una di esse, sarà un fatto automatico, ma il sapere come avviene questo processo può risultare utile in molte circostanze pratiche.

SETENV

Con questo comando è possibile impostare o azzerare il contenuto di una variabile. Impartirlo da solo non produce altro effetto che un errore *Bad Args*.
In effetti, per assegnare un contenuto ad una variabile, non sempre è utile adoperare Setenv. Considerata la natura della variabile, che altri non è che un file, in certe occasioni è preferibile adoperare un semplice **reindirizzamento**. Un esempio banale: si apra una finestra shell dal disco Workbench (che dovrà essere stato adoperato per il boot!), e si impartisca al suo interno `Setenv Var Devs` (e Return). Si sarà così creata una variabile di nome **Var**, il cui contenuto sarà la stringa **"devs"**.
L'uso di questa variabile sarà utile nell'ambito di batch files che ne testino eventualmente il contenuto con qualcosa come **"If $var eq devs..."**, ma in modo diretto se ne può constatare l'esistenza con qualcosa come **`Dir <Env:Var`**.
Si otterrà la visualizzazione della directory devs, se accessibile. Giusto come prova, si digiti **`Type Env:var`**. Si constaterà come il file Var è un banale file di testo, il cui contenuto è "devs".
Proprio per questa caratteristica, lo stesso effetto lo si può raggiungere (p. es.) con **`Echo >env:var Pippo`**. In questo caso, la variabile Var assumerà il nuovo contenuto "pippo". Per verificarlo, oltre che Type, si può anche adoperare **`Getenv Var`** (e return): si vedrà apparire a video l'immancabile Pippo (non lui di persona, comunque...).

SETENV variabile stringa

Il nome da assegnare alla variabile. Trattandosi, in realtà, di un file, risponde alle regole generali del dos, ovvero indipendenza dal maiuscolo / minuscolo, e numero di caratteri fino ad un massimo di **30**.
Adoperando Setenv con questo solo parametro, senza farlo seguire dalla specifica "stringa", il contenuto della variabile viene invece azzerato. Quindi, se con Setenv Var Pippo si assegna "pippo" alla variabile Var, con **`Setenv Var`** questa non conterrà più alcuna stringa.
Non si dimentichi che, qualora si adoperi un nome contenente degli spazi, questo andrà racchiuso tra **virgolette** (si veda il numero scorso della rivista per tutti i dettagli), e che lo stesso andrà fatto per eventualmente leggere la variabile con Getenv (leggi: sconsigliabile).

Il contenuto della variabile. Come già accennato, questo diventerà comunque un testo Ascii, ed in quanto tale non ha rilevanza il fatto che sia di contenuto numerico o alfabetico. Se si ha la necessità di adoperare dei valori numerici, questi potranno comunque essere vagliati da **If** associato all'opzione **Val** ed essere interpretati correttamente anche da **Eval**, se risultasse necessario sottoporli a calcoli matematici (si veda l'esempio).
Come nel caso del nome, anche questo parametro va tacchiuso tra doppi apici se contiene degli spazi.

La pratica: Multiformat

Per capire a fondo come implementare l'uso di variabili all'interno di batch file, niente di meglio che un esempio pratico, che riassume tra l'altro molte tecniche già affrontate e descritte nel corso degli appuntamenti con *AmigaFacile*.
Un esempio però non astratto, ma che ci fornisca di una utility tutta Dos che può risultare vantaggiosa in più di un'occasione. E, naturalemnte, che vada bene per tutti, anche per chi ha solo un **Amiga 500 senza drive esterno**.
Questa applicazione la chiameremo **MultiFormat**, in assonanza al compito che si prefigge: formattare automaticamente una serie di dischetti, il cui numero dipende solo da una scelta personale. Il tutto senza dover fare altro che infilarli nel drive (beh, uno alla volta, questo non lo si può evitare...) e premere return.

GETENV

Con questo comando, si ottiene molto semplicemente la visualizzazione del contenuto della variabile specificata come parametro. In pratica: se dopo un `Setenv Var Pippo` si impartisce `Getenv Var`, verrà visualizzata la stringa **Pippo**.

E' possibile inoltre adoperare la redirezione per copiare lo stesso contenuto in un'altra variabile. Proseguendo nel nostro esempio (lo si metta in pratica da shell), con `Getenv >newvar Var` si creerà una seconda variabile di nome **Newvar**, il cui contenuto sarà sempre "Pippo".
Come si potrà constatare nell'esempio di queste pagine, Getenv trova in effetti scarso utilizzo nell'ambito dei batch file, che restano l'obiettivo principe nell'uso delle variabili di ambiente.

GETENV	variabile

Vediamo anzitutto il suo funzionamento, passeremo poi ad una analisi dei suoi punti "caldi", incentrati appunto sull'uso di **variabili ambiente**.
Anzitutto, è necessario copiare il file mostrato in queste pagine con un editor Ascii (oppure **Ed**), e memorizzarlo di preferenza nella directory **S** del disco Workbench se si pensa di adoperarlo usando il comando **Execute**. In alternativa, ormai dovrebbe essere acquisito, si può adoperare il comando **Protect MultiFormat +S**, con la possibilità di mandarlo in esecuzione senza adoperare Execute: in questo caso, sarebbe più comodo memorizzarlo nella directory **C**, in modo da risultare sempre accessibile semplicemente invocandone il nome.

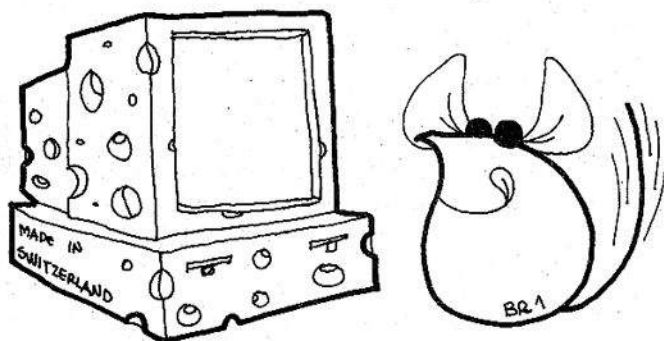
La sua sintassi d'uso sarà: `MultiFormat xx` (oppure `Execute Multiformat xx`), con **XX** che dovrà specificare il numero di dischetti da formattare. Inutile aggiungere che occorre tener presente l'eventuale percorso del file nell'invocarlo (p. es. `Execute Ram:Multiformat xx` se è memorizzato in ram disk), mentre si dà per scontato che Amiga sia stato lanciato con il disco Workbench, o con una a sua copia. In caso contrario, si renderebbe necessario provvedere da sé ad assegnare **Env:** come prima descritto. Se, per errore, si digitasse il comando senza alcun parametro, questo si limiterebbe a segnalarlo a video, arrestando la sua esecuzione.
Prima di ogni altra cosa, viene richiesto se si desidera una fantomatica **procedura turbo**. Cosa significa? Semplice: i batch file sono di solito piuttosto lenti, in quanto tutti i comandi che li compongono devono essere caricati da disco. Per di più, un'applicazione di questo tipo diventerebbe impraticabile per chi dispone di un solo drive. Il nostro batch, per ovviare all'inconveniente, rende residenti tutti i comandi in esso contenuti, garantendo una velocità di tutto rispetto, e soprattutto consentendo anche a chi possiede un solo drive di usufruirne facilmente.
Due sole annotazioni: l'unico comando a non essere reso residente, ma solo trasferito in Ram Disk, è proprio **Format**, in quanto questo non è "rientrante" (manca infatti del bit "pure" settato: si veda il manuale alla voce Resident). Inoltre nel batch file è stata inserita la richiesta se si desidera la procedura turbo per un motivo molto semplice: dopo la conclusione delle operazioni, i comandi resi residenti rimarranno tali, a meno che non si preferisca aggiungere al batch una serie di comandi `Resident xxx Remove` per eliminarli dalla memoria. Se si dovesse riutilizzare MultiFormat nella stessa sessione di lavoro, non sarebbe dunque necessario ripetere questa operazione, basterebbe rispondere **N** (no) alla richiesta sul "turbo", continuando però ad usufruire dei suoi benefici.
Dopo questa premessa, entriamo nel vivo dell'argomento, raccomandando un attento esame del batch e dei suoi commenti, che delimitano le varie fasi operative.
Nella fase che segue la serie di comandi Resident, si nota subito l'uso di **Setenv** per assegnare valore **0** oppure **1** alla variabile **Flag**, a seconda che si disponga di un solo drive o anche del drive esterno. L'effetto di questo flag lo si può vedere più avanti, nella sezione che si occupa in senso stretto della formattazione, ove una condizione `If Val $flag eq 0` (oppure `eq 1`) testa appunto il valore della variabile, e a seconda dell'esito adopera una sintassi del comando Format rivolta al drive **df0:** oppure al drive **df1:**.
In pratica, il solo uso di una banale variabile aggiunge una notevole flessibilità al batch, che può così essere adoperato in diverse configurazioni hardware. Si noti, già che ci siamo, come dopo Format sia presente una doppia redirezione (**>nil: <nil:**): la prima serve solo

per non mostrare l'output del comando, mentre la seconda aggiungerà l'equivalente di un Return, evitando una doppia pressione dello stesso da parte dell'utente. I floppy assumeranno tutti "**X**" come nome, eventualmente poi modificabile, ma la cosa non porrà problemi di sorta (si veda descrizione di Format in Amigafacile n. 76).
Torniamo di nuovo indietro nel batch, per trovare le due istruzioni **.bra** e **.ket**, che servono a modificare i caratteri di default "**<**" e "**>**" per delimitare il parametro numerico immesso al lancio del comando (vengono sostutuiti da "[" e "]"). Ciò si rende necessario in quanto i simboli di redirezione presenti nel file-comandi verrebbero interpretati erroneamente dal dos.
A questo punto, ha inizio un vero e proprio "loop" (o ciclo), sullo stile del For...Next del basic, per intenderci. Per realizzarlo, ecco di nuovo in azione Env:.
Prima di tutto si crea una variabile contenente il numero di floppy da formattare, adoperando Echo e una redirezione verso **Env:num**. In pratica, la variabile **Num** conterrà il numero di dischi da formattare.
Si stampa a questo punto l'invito ad inserire il disco (si noti la tecnica adoperata per stampare df0: oppure df1: a seconda del caso), e "parte" la prima formattazione. Dopodiché, sarà necessario diminuire di una unità il valore della variabile Num, e controllare se sia giunta a zero: in caso negativo, un salto all'indietro (**skip ... back**) alla label "loop" farà ricominciare la procedura.
Anche in queste fasi, le variabili la fanno da padrone. Per il controllo (che simula il Next del basic), è sufficiente un banale `If val $Num gt 0`, che sta per *"Se il valore della variabile Num è superiore a 0, allora..."*.

Più complessa invece l'operazione per sottrarre una unità al valore di $Num. Schematicamente, l'operazione si svolge così:

1) Si effettua una sottrazione (di 1 unità) dalla variabile Num adoperando **Eval** (si veda numero scorso di AmigaFacile), ed il risultato lo si deposita in una variabile intermedia **Num2**.
2) Si copia il contenuto di Num2 nella variabile **Num**.

Macchinoso... ma è l'unico modo; Eval normalmente si limita a mostrare il risultato, senza modificare i due termini dell'operazione.
Ancora più macchinosa la sintassi dell'operazione, che sfrutta l'uso dei cosiddetti "template" ottenibili mediante il punto interrogativo. Senza troppo approfondire (se ne riparlerà nei prossimi appuntamenti), basti sapere che vi sono due metodi per assegnare dei parametri ad un comando quando lo si fa seguire da "**?**": digitarli direttamente, e nel giusto ordine, è quello più noto.
Ma si può anche (come avviene nel batch) citare l'esatta dizione del template (per conoscerla, basta impartire p. es. Eval ?), e far seguire al simbolo di uguale il parametro da inserire. Così, nel nostro Multiformat, il primo termine di Eval viene prelevato (tramite la redirezione "<") dalla variabile Num, il secondo si specifica (con value2=1) espressamente, ed infine si comunica ad Eval che l'operazione è una sottrazione (**op=-**). L'esito sarà diretto verso env:num2, ovvero la variabile Num2, adoperando la keyword TO.
Tutto questo, direte, solo per ottenere un banale Num=Num-1!
Ebbene sì, il dos non è proprio un linguaggio di poche parole. Come darvi torto...?

In termini di praticità, il nuovo comando Multiformat mostrerà ad ogni formattazione il numero di dischi ancora da trattare, e non si dovrà fare altro che inserire i floppy nel drive, premere Return, e attendere (...). Se si dovessero reinizializzare dei vecchi dischi già in formato Amiga, si può velocizzare il tutto aggiungendo **Quick** alla fine delle righe che contengono il comando **Ram:format**. Stesso discorso per un eventuale **Noicons**, che eviterà la creazione della directory **Trashcan** nei dischi appena formattati.
Capito il funzionamento del batch, su questa impronta si potrà realizzare una marea di applicazioni, o anche modificare lo stesso MultiFormat per adattarlo ai propri gusti.
Buon Dos!

```
.key num
;----------------------------------
resident c:echo
if <num>x eq x
 echo "*ecDevi specificare quanti"
 echo "dischi vuoi formattare!"
 echo "Sintassi:"
 echo "Multiform xx (xx=numero dischi)"
 echo " "
 quit
endif
;---------------------------------
if val <num> ge 1
 echo "*ecVuoi la procedura turbo?"
 echo "(indispensabile con solo 1 drive!)"
 echo " "
 ask "Y = Yes  N = NO   "
 if warn
  echo "*ecAttendi..."
  resident c:resident
  resident c:copy
  resident c:if
  resident c:endif
  resident c:else
  resident c:quit
  resident c:ask
  resident c:assign
  resident c:setenv
  resident c:lab
  resident c:type
  resident c:eval
  resident c:skip
 endif
;-------------------------------- --
 copy sys:system/format ram:
 ask "*ecHai un solo drive? (Y/N)"
;----------------------------------
 if warn
  echo >ram:1d "nel drive interno"
  assign fil: ram:1d
  setenv flag 0
 else
  echo >ram:2d "nel drive Df1:"
  assign fil: ram:2d
  setenv flag 1
 endif
;-----------------------------
.bra [       ;Non indentare
.ket ]       ;queste due righe!
;-----------------------------
 echo >env:num [num]
 lab loop
 echo "*ec"
 type env:num
 echo " "
 echo "Inserisci un disco"
```

Intercetta l'eventuale assenza del parametro indicante il numero di dischi da formattare.

Rende residenti tutti i comandi del batch, per velocizzare le operazioni e consentirne l'uso anche ai possessori di un solo drive.

Prepara variabile-flag e parametro di format (df0: oppure df1:) in base alla configurazione hardware.

Modifica i caratteri di riconoscimento del Dos. Non indentare queste due righe, altrimenti si otterrebbe un "Keyword too long"!

Inizializza variabile/contatore del ciclo e inizia loop.

```
 type fil:
 ask "e premi RETURN"
 echo " "
 echo "Attendi...."
;-----------------------------------
 if val $flag eq 0
  ram:format >nil: <nil: drive df0: name X
 endif
 if val $flag eq 1
  ram:format >nil: <nil: drive df1: name X
 endif
;-------------------------------------
 eval <env:num >nil: to env:num2 value2=1 op=- ?
 type >env:num env:num2
;----------------------------------------
 if val $num gt 0
  skip loop back
 endif
;----------------------------------------
 assign fil:
 echo "*ecFATTO!"
 echo " "
else
 echo "Parametro non valido!"
endif
```

Formatta adoperando il drive interno oppure Df1:, a seconda del contenuto della variabile Flag. Aggiungere eventualmente Noicons e/o Quick in coda alle righe di Format.

Aggiorna il contatore del loop, sottraendo una unità alla variabile Num.

Stessa funzione del Next del basic: controlla la variabile- contatore, e se ha non raggiunto lo zero continua il loop.

Resetta assegnazione di fil: e conclude le operazioni. L'Else va riferito al primo If Val <num> Ge 1, che include tutta la routine. Diventa attivo se per esempio si è dato come parametro un numero negativo.

Già pubblicati

A partire dal n. 75 della rivista si è cominciato ad esaminare in profondità il Dos 1.3 di Amiga, rivolgendosi soprattutto (ma non solo) a chi si avvicina per la prima volta a questo computer. Chiaro che, con il succedersi degli appuntamenti, l'integrazione tra quanto trattato in precedenza e gli argomenti affrontati in queste pagine si fa sempre più stretta, anche se viene volutamente mantenuta la maggiore autonomia possibile. Per quei lettori che inizino solo ora a seguire la nostra rivista, è consigliabile (anche se non indispensabile) procurarsi tutti i numeri a partire dal già citato 75: si disporrà così di un completo compendio da affiancare alla consultazione del manuale, spesso fin troppo succinto nelle sue indicazioni. Ecco comunque un elenco di quanto già pubblicato nelle pagine della rubrica Amigafacile.

n.75 (Giugno 1990)
Assign
Copy
Date
Dir
Install
Path
Search
Sort

n.76 (Lug/Ago 1990)
Caratteri Speciali
Delete
Format
Protect
Rename

n.77 (Settembre 1990)
Execute
Direttive Batch
If
Skip
lab
Quit

n.78 (Ottobre 1990)
Cd
Ed
Break

n.79 (Novembre 1990)
Which
Newshell E Newcli
Ser: Par: E Prt:
Con: E Newcon:
Nil: E Raw:

n.80 (Dicembre 1990)
Setmap
Iconx
List

n.81 (Gen/Feb 1991)
Avail
Alias
Join

n.82 (Marzo 1991)
Failat
Eval

N.B.
Le parole chiave di ogni comando sono rappresentate in maiuscolo e vanno (eventualmente) adoperate così come sono. In minuscolo sono invece riprodotti i parametri che vanno ridefiniti dall'utente.

di Luigi Callegari

LA GESTIONE DELLA MEMORIA

Proviamo a "giocare" con la memoria Ram di Amiga

Per riservare una certa quantità di memoria, ovvero "allocare RAM", un programma in linguaggio C può utilizzare fondamentalmente tre modi: usare le funzioni di libreria standard, usare le funzioni di Exec oppure definire matrici di variabili (di carattere, tipicamente).

I metodi sono simili, ma non identici. Innanzitutto, la definizione di una variabile rappresenta una allocazione di tipo "statico", fatta cioè una volta per tutte dal compilatore durante la codificazione del testo sorgente. Il vantaggio è che durante l'esecuzione non viene perso tempo dal programma per allocare la memoria; lo svantaggio è che non è possibile allocare quantità variabili di memoria, come potrebbe richiedere un nostro programma allo scopo di avere maggiore efficienza.

Le funzioni di gestione della memoria sono di due tipi complementari: per l'**allocazione** e per la **deallocazione** di aree. Nel caso della libreria di Exec, che tratteremo prossimamente su queste pagine, possiamo anche specificare che tipo di memoria di Amiga vogliamo allocare: Chip Ram, Fast Ram oppure Public Ram.

L'esigenza deriva dalla necessità peculiare, in campo Amiga, di memorizzare dati entro una certa sequenza di locazioni di Ram (i primi 512 K o 1024 K, secondo il modello di ECS montati nel proprio computer).

Usando invece delle funzioni standard di libreria, come **malloc()**, non è possibile specificare quale tipo di memoria si desidera; esattamente come quando si allocano staticamente delle variabili.

Funzioni di livello 3

Le funzioni di libreria standard Ansi / Unix, comuni perciò anche ai compilatori standard per Ms - Dos (Zortech C, Microsoft C e Turbo C, ad esempio), non sono molte. Si dividono in tre livelli, diversi per alcune caratteristiche ma abbastanza simili concettualmente, come abbiamo visto nel numero scorso per i vari livelli delle funzioni di gestione dei file.

Le funzioni di più alto livello sono riportate in figura 1: **calloc(), malloc(), realloc()** e **free()**.

La più usata è malloc(), che accetta un parametro singolo come argomento indicante quanti bytes di memoria riservare e restituisce un puntatore che l'utente può forzare alla dimensione voluta, oppure un NULL (0) se non è stato possibile, per qualunque motivo, allocare la memoria.

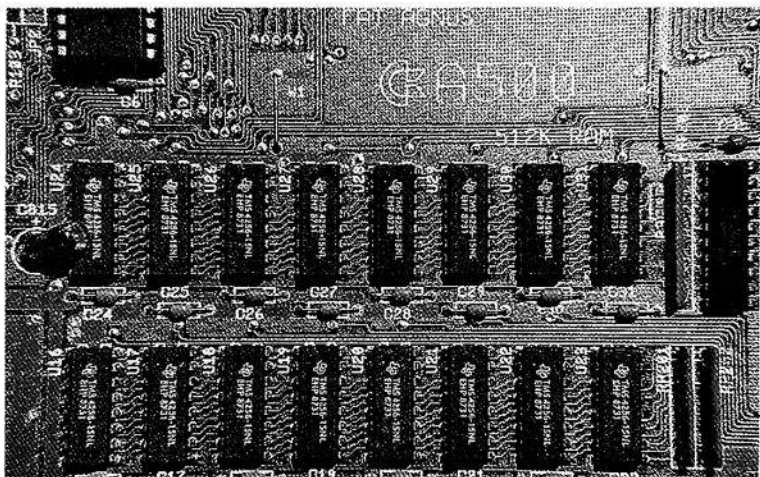
Ad esempio, supponendo di volere allocare 500 byte di memoria da usare per memorizzare numeri interi, useremo...

```
int *pint;
pint = (int *)malloc( 500 );
```

...in cui la variabile **pint** contiene, dopo l'esecuzione, l'indirizzo di partenza dell'area allocata, oppure NULL se non è stata possibile l'allocazione. E' bene inserire sempre una verifica del valore restituito, altrimenti il nostro programma potrebbe ritrovarsi a scrivere nelle prime locazioni di memoria (se NULL vale zero), provocando una Guru Meditation.

Spesso però capita di dover allocare una certa quantità di memoria per contenere elementi la cui dimensione viene calcolata dal compilatore. Questo è il caso di utilizzo dell'operatore standard del C, sizeof():

```
struct nodo *p;
p = (struct nodo *)malloc(size-
of(struct nodo));
```

In questo caso la variabile **P** punterebbe ad un numero di locazioni di memoria

Multitasking e gestione della memoria

Chi proviene da ambienti di programmazione più semplici, come ad esempio il rassicurante C/64, può trovare strani i concetti di "allocazione di memoria", comuni a tutti gli elaboratori di classe superiore (tra cui Amiga).

Programmando computer come il **C/64** o lo **Spectrum** in Assembler o in Basic, la memoria era proprio come una lavagna sulla quale si poteva scrivere liberamente, fatte salve alcune zone ben definite, usate dal sistema per conservare i suoi dati di lavoro (mappa schermo e periferiche, variabili di sistema, eccetera).

Amiga è una macchina dotata di un sistema operativo evolutissimo e multitasking. Per consentire a più programmi, ovvero a più **task**, di funzionare contemporaneamente in uno stesso computer, è molto importante che sia impedito di disturbarsi a vicenda, ad esempio consentendo loro di usare incidentalmente le stesse aree di memoria.

L'allocazione di memoria, dunque, è un metodo per segnalare ad **Exec**, il cuore del sistema operativo di Amiga, che certe aree di memoria sono usate da un programma e, quindi, non possono essere utilizzate da altri. La memoria da usare deve essere richiesta a Exec, che restituisce sicuramente l'indirizzo di un'area libera, dopo avere consultato le sue tabelle interne, dove conserva l'indirizzo di tutte le aree già riservate indicativamente.

L'onere della gestione delle aree di memoria allocate e della loro "restituzione" al sistema (ovvero della segnalazione ad Exec che sono riutilizzabili perchè non più usate da un task, operazione detta **deallocazione**) è compito del programmatore, che deve usare in linguaggio C le funzioni di cui si parla in queste pagine.

sufficiente a contenere una struttura *"nodo"* precedentemente definita nel file. Il puntatore viene memorizzato in p.

La funzione calloc() funziona in modo molto simile a malloc(), ma provvede ad azzerare anche la memoria che alloca e consente di specificare, come parametri, il numero di elementi che si desidera memorizzare e la dimensione in bytes di ciascuno di essi, calcolata con il solito sizeof():

```
double *f;
f=(double *)calloc(50,sizeof(double));
```

In tal modo si ottiene, in **Fd**, un puntatore ad un'area di memoria azzerata, sufficente a contenere 50 elementi in formato double.

Considerazioni importanti

E' buona regola usare **sizeof()** nei propri programmi per fare calcolare durante la compilazione la dimensione effettiva di un oggetto, in quanto ciascun compilatore può usare standard interni differenti per memorizzare numeri e dati (specialmente quando il programma deve essere compilato e fatto girare su elaboratori differenti). Il risultato dimensionale di sizeof() sarà sempre corretto in funzione dell'ambiente ospite del programma eseguito.

Anche la funzione **calloc()** restituisce un puntatore nullo se non è riuscita ad allocare la memoria richiesta. La funzione **free()** rilascia una zona di memoria allocata con malloc(), calloc() o realloc(), rendendola cioè nuovamente disponibile al sistema per altri usi.

Ovviamente, è indispensabile restituire correttamente al sistema operativo le aree di memoria non più necessarie, possibilmente non appena terminato di usarle. In effetti, i "linker" dei compilatori di Amiga inseriscono sempre un modulo di inizializzazione che provvede a liberare tutte le aree di memoria riservate con le funzioni di allocazione, analogamente a quanto visto per i file aperti con **fopen()**. Comunque, è buona regola indicare sempre esplicitamente un free() per ogni allocazione fatta. L'argomento di free() è il puntatore restituito da una delle funzioni di allocazione:

```
free( puntatore );
```

L'ultima funzione standard di allocazione della memoria è realloc(), che consente di allocare un nuovo blocco di memoria ricopiandovi i dati di uno vecchio, e deallocando quest'ultimo. Ad esempio:

```
char *mp, *vp;
vp = (char *)malloc( 500 );
mp = (char *)realloc( vp, 700 );
```

In questo caso si ottiene l'allocazione normale di **500** byte con puntatore memorizzato in **Vp.** Con la terza linea, invece, si provvede ad allocare **700** byte di memoria, ricopiarvi i 500 byte (con i dati memorizzati da funzioni intermedie tra la seconda e la terza linea del nostro esempio, nella realtà), riservati precedentemente ed indicati dal puntatore Vp, memorizzando in Mp il nuovo puntatore al

Funzioni di gestione memoria di livello 3

Ecco i formati canonici e le descrizioni delle funzioni di gestione memoria di livello 3. Per usare queste funzioni è necessario includere il file **stdlib.h** che comprende le necessarie dichiarazioni e prototipi:

```
void *malloc( unsigned n );
```

Restituisce un puntatore nullo ad un'area di memoria ampia almeno "n" bytes.

```
char *realloc( void *b, unsigned n );
```

Alloca un nuovo blocco ampio "n" byte, poi copia il vecchio blocco "b" sul nuovo blocco "nb" e rilascia il blocco precedentemente allocato. Se è inferiore alla dimensione del blocco precedente, copia solo i primi "n" bytes.

```
void *calloc( unsigned nelt, unsigned esize );
```

Usa la funzione malloc() che ricava un blocco di dimensione in byte pari a (nelt * esize) e ne restituisce l'indirizzo come puntatore nullo.

```
int free( void *b );
```

Rilascia un blocco di memoria, del quale viene indicato l'indirizzo iniziale precedentemente ottenuto con calloc(), malloc() o realloc().

Figura 1

```
/* Ricerca errori in File ASCII
   Per AMIGADOS SAS/C V5.10
****************************** */
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <stdio.h>
#ifdef LATTICE
# include <proto/dos.h>
#else
# include <functions.h>
#endif
#define  MAXLENW  20 /* Massima lunghezza parola
+ 1 */
#define  MINLENW  3  /* Minima lunghezza */
#define  VOCALE   1  /* Valore per vocale */
#define  CONSON   3  /* Valore per consonante */
#define isvoc(c)
((c)=='e'||(c)=='i'||(c)=='a'||(c)=='o'||(c)=='u')

int voci = 1;      /* Numero voci stampate */
int nasc = 0;      /* Numero caratteri non ASCII
*/
int par1 = 0;      /* Contatore parentesi tonde */
int par2 = 0;      /* Contatore parentesi quadre
*/
int par3 = 0;      /* Contatore parentesi graffe
*/
/* Struttura conteggio parole */
struct nodo {             /* Struttura di un nodo
*/
  char        *parola;  /* Puntatore alla parola
*/
  unsigned    conta;    /* Conteggio ricorrenze
nel file */
  struct nodo *min;     /* Puntatore a nodo pre-
cedente */
  struct nodo *mag;     /* Puntatore a nodo suc-
cessivo */
};
/* Funzione di gestione errori */
void gerror( char *s, int n )
{
     printf("\nErrore %d: %s\n",n,s);
     exit( 1L );
}
/* Alloca memoria per un nuovo nodo
*******************/
struct nodo *palloc( void )
{
     return( (struct nodo *)malloc(sizeof(struct
nodo)));
}
/* Alloca memoria e salva una parola nuova
************/
char *savestr( char *s )
{
     char *p;
     p = (char *) malloc( strlen( s+1 ) );
     if ( p == NULL )
        return( p );
     else
        return( (char *) strcpy( p, s ) );
}
/* Stampa ricorsivamente l'albero
**********************/
void displyalbero( struct nodo *p )
{
     if ( p != NULL )
     {
         displyalbero( p -> min );
         if ( p -> conta == 1 )
         {
             printf( "%19s", p -> parola );
             if ( !( voci++ % 4) )
               printf("\n");
         }
         displyalbero( p -> mag );
     }
}
/* Inserisce un nuovo elemento nell'albero */
struct nodo *insert( struct nodo *p, char *w )
{
     int c;
     if ( p == NULL )
     { /* Nuova parola */
         if ( ( p = palloc() ) == NULL )
              gerror("Non riesco ad allocare RAM
per il nodo!",1);
         p -> parola = savestr( w );
         p -> conta  = 1;
         p -> min = p -> mag = NULL;
     }
     else if (( c = strcmp(w,p->parola)) == 0 )
/* Parola ripetuta */
         p -> conta++;                      /* In-
crementa conto */
     else if ( c < 0 )
         p -> min = insert( p -> min, w ); /* In-
serisce prima */
     else
         p -> mag = insert( p -> mag, w ); /* In-
serisce dopo */
     return(p);
}
```

Il listato di esempio (prima parte)

E' opportuno ricordare che alcune istruzioni del programma, che apparentemente sembrano appartenere a più righe, vanno invece digitate su un'unica linea-video; esigenze di impaginazione, infatti, talvolta impongono un "formato" insolito ai listati pubblicati sulla nostra rivista.

NEWEL® srl

computers ed accessori

20155 MILANO via Mac Mahon, 75

NEGOZIO	tel.0 2 / 3 2 3 4 9 2
UFFICI	tel0 2 / 3 2 7 0 2 2 6
FAX 24h	tel0 2 / 3 3 0 0 0 0 3 5
UFFICIO SPEDIZIONI	tel.0 2 / 3 3 0 0 0 0 3 6

UNICA SEDE IN ITALIA

VENDITA ANCHER PER CORRISPONDENZA IN TUTTA ITALIA

EVASIONE ORDINI NELLE 24 ORE SUCCESSIVE ALL'ORDINE

SUPER 64 EMULATOR
Ultima versione del famoso emulatore C64, completo di interfaccia hardware per la connessione con le periferiche del 64
L. 29.000

MIDIMASTER
Interfaccia midi professionale con due ingressi, 1 porta passante (TRUH) ed 3 uscite, progettata da esperti MIDI, tutti i segnali standard, altissima qualità disponibile per A500/1000/2000/3000.
L. 79.000

MOUSE SELECTOR
Utilissimo permette di collegare contemporaneamente il mouse ed il joystick e selezionare tramite interuttore quello desiderato, senza dover ogni volta sconnettere e rischiare di danneggiare il computer. **L. 29.000**

JOYSTICK IN OFFERTA
SUPER JOYSTICK
Dagli USA, il robustissimo "Super Joystick anima in metallo, sei microswich di precisione, bottone di fire sull'impugnatura e sulla base, autofire disinseribile tramite swich, il tutto in un nuovissimo design nero con ventose.
L. 35.000

JOYSTICK "CLOCHE"
per simulatori di volo
Altra novità dagli Stati Uniti, questo rivoluzionari è destinato ad una fascia di utenti ben precisa, gli appassionati dei simulatori di volo. Vi sembrerà di guidare un vero aereo, una vera "CLOCHE" con i stessi comandi di un aereo, il tutto dotato di microswich ed ventose, il divertimento è assicurato se lo userete con programmi tipo: Flight Simulator, Mig 29, Falcon, F16 Combat Pilot, F19 Stealth Fighter e molti altri. **L. 69.000**

DIGITALIZZATORI VIDEO
DIGI VIEW 4,0 GOLD
Digitalizzatore professionale il più affidabile per risultati incredibili, a colori in altissima risoluzione, filtratura manuale, completo di istruzioni e software originale **L. 290.000**

VIDEON 3,0
Nuovo digitalizzatore professionale a colori ara a 32000 colori, con SUPER-VHS, nuovo software di gestione per creare sempre immagini più incantevoli lavoro con immagine ferma per almeno 13 secondi, completo di cavi e istruzioni in italiano **IN OFFERTA**

MOUSE PAD - TAPPETINO ANTISTATICO PER MOUSE
Utilissimo tappetino antistatico per mouse, indispensabile per un buon utilizzo del mouse, evita il contatto diretto con superfici non idonee per un corretto uso del mouse. **L. 15.000**

TAVOLA GRAFICAPROFESSIONALE "PODSCAT" PER AMIGA
Tavoletta grafica professionale per tutti gli amiga, permette di disegnare con una penna a sfera su una superficie, data la sua precisione è ideale per disegno tecnico, grafico e pittorico, corredato di driver ed istruzioni semplice da usare. **L. 649.000**

CONTROLLER GVP SERIE II PER A2000
Il nuovissimo controller della Great Valley Production per Harddisk con le nuove FASTROM. La scheda è espandibile con moduli Simm fino ad 8 Mb di ram, installazione semplicissima ed velocissima **L. 490.000**
OGNI MODULO DI RAM DA 2MB
L. 275.000

AMIGA PC AT EMULATOR
MANUALE IN ITALIANO
ATONCEPC 286 EMULATOR AT
AMIGA 500 MULTITASKING
Lire 390.000
ORA ANCHE PER A2000

• ESPANSIONI DI MEMORIA PER AMIGA 500, 1000 E 2000
Costruite con i migliori materiali, le nuove espansioni di memoria dell'ultima generazione usano i nuovissimi chip da 1 mbit che sono notevolmente più veloci, autoconfiguranti, slim line, e con 1 anno di garanzia!

• 512k per amiga 500	L. 85.000
• 512k per amiga 500 + clock	L. 110.000
• 1,5 mb per amiga 500 + clock	L. 240.000
• 4 mb per amiga 500 +clock	L. 590.000
• 2 mb per amiga 1000	L. 450.000
• 2 mb espandibile a 8 per amiga 2000 + clock	L. 390.000

UTILITY DISK DEL VALORE DI L. 50.000 IN OMAGGIO

DRIVE INT/ESTERNI PER TUTTI GLI AMIGA ALTA QUALITÀ
GARANZIA 12 MESI

• DRIVE INTERNO PER AMIGA 500	L. 159.000
• DRIVE ESTERNO PER AMIGA 500/1000 PASS. + DISCONNECT	L. 159.000
• DRIVE INTERNO PER AMIGA 2000 (COMPLETO)	L. 149.000
• DRIVE ESTERNO PER AMIGA 500/1000/2000 DA 5"1/4	L. 249.000

UTILITY DISK DEL VALORE DI L. 50.000 IN OMAGGIO

BOOTSELECTOR L. 19.000
Trasorma il drive esterno in DFO: (interno) utile per evitare l'usura eccessiva del drive interno, e risolvendo inoltre problemi di compatibilità con il drive originale, Kit di semplicissima installazione.

KICKSTAR 1.2 & 1.3 ROM L. 99.000
Scheda da montare semplicissimamente all'interno del vostro amiga 500/2000 e vi permette di avere a disposizione i due sistemi operativi 1.2 per la compatibilità con tutti i giochi, 1.3 per le nuove espansioni l'hardisk ecc. Indispensabile!!!, (non necessita di saldature) specificare versione richiesta.

ESPANSIONE A501 (ORIGINALE COMMODORE)
SUPEROFFERTA L. 130.000
Porta il tuo amiga a 1MB di ram + orologio in tempo reale e calendario.

AMTRAC - TRACBALL AMIGA
Grandiosa novità, finalmente disponibile per Amiga il noto tracball si sostituisce al mouse esegue alla perfezione le medesime funzioni, ed oltre ad avere una sensibilità e precisione indubbiamente migliore, risolve moltissimi problemi di spazio bastano 20 cm.e dato che non bisogna spostarlo basta sfiorare la sfera, utile, semplice, divertente e sopratutto molto preciso.
IN OFFERTA L. 89.000

ACTION REPLAY 2 NOVITÀ
(disponibile anche per AMIGA 2000)
la prima cartuccia rivoluzionaria multifunzioni per amiga 500/1000 con opzioni di freeze: permette di sproteggere la maggior parte dei programmi in commercio (consentendoti di creare giochi di sicurezza per uso personale, inoltre permette di creare giochi trainer, (vite infinite ecc.), permette di bloccare un gioco in qualsiasi momento dal medesimo posto, salva una qualsiasi videata (disegno, testo) su disco, consentendoti una facile hardcopy anche su stampante, funzione moviola (rallenta programmi e giochi), potente virus-detector, sprinteeditor, oltre che ad un monitor straordinario per il linguaggio macchina, questo e molto, molto di più, ti aspetta in amiga action replay!!!
il tutto ad un prezzo eccezionale!
versione originale con manuale in italiano
L. 169.000

INTERFACCIA 4 JOYSTICK
Permette di collegare contemporaneamente 4 joystick all'amiga e quindi di giocare in 4 contemporaneamente a giochi tipo CALCIO, PALLAVOLO ecc.
DISPONIBILE !!! **L. 29.000**

TURBO ACCELERATOR CARD PER AMIGA 500/1000/2000
Funziona su tutti gli amiga, semplice installazione, non necessita di saldature, 100% compatibile, velocità selezionabile tra 7 e 14 Mhz, 32 kb di ram statica ad alta velocità, di cui 16 come cache memory.
Il più veloce acceleratore senza memoria a 32 bit. STRAORDINARIO!!!
L. 450.000

fish-disk aggiornati al n° 400 tutti!

richiedi il nostro nuovo catalogo gratuitamente specificando il computer posseduto!

Tutte le ultime novità software, originali, importazione diretta!!!
Disponibile in magazzino l'intera gamma professionale Amiga 3000

TUTTI I NOSTRI PRODOTTI SONO COPERTI DA GARANZIA DI 12 MESI

TUTTI I PREZZI SONO IVA 19% COMPRESA

COMMODORE AMIGA 500 E 2000 AI PREZZI PIU' BASSI D'ITALIA. TELEFONARE

```
/* Legge una stringa accettabile dal file ed ese-
gue controlli */
char *getword( char *s, int n, FILE *infile)
{
     char c, *cs;

     cs = s;
     while ( --n > 0 && ( c = getc( infile ) )
!= EOF )
     {
     if ( ( c < 32 && c != 10 ) || c > 127 )
         ++nasc; /* Carattere non ASCII */
     switch( c )
     {
         case '(': ++par1; break;
         case ')': --par1; break;
         case '[': ++par2; break;
         case ']': --par2; break;
         case '{': ++par3; break;
         case '}': --par3; break;
         default:          break;
     }
     if ( ( c = tolower( c ) ) < 'a' || ( c ) >
'z' )
         break;
     else
         *cs++ = c;
 }
 *cs = NULL;
 return( ( c == EOF && cs == s ) ? NULL : s );
}
/* Verifica correttezza di una parola (TRUE/FAL-
SE) */
BOOL checkw( char *w )
{
  int i, c, flag1 = 0, flag2 = 0;
  for ( i = 0 ; i < strlen( w ) - 1 ; i++ ) {
      if ( ( c = isvoc(*(w+i)) ) ==
isvoc(*(w+i+1)) ) {
          if ( c == TRUE ) { /* Doppia vocale */
             if ( *(w+i) == *(w+i+1) )
                return( FALSE ); /* Sono uguali */
             else if ( ++flag2 == 2 )
                return( FALSE ); /* 3 vocali  */
             }
          else if ( ++flag1 == 2 )
             return( FALSE ); /* Tripla consonante
*/
          }
      else flag1 = flag2 = 0;
      } /* fine del for() */
      return( TRUE );
}
void main( int argc, char *argv[])
{
  struct nodo *first = NULL;
  char item[MAXLENW+1];
  FILE *infile;
  LONG countword = 0;
  if ( argc != 2 )
      gerror( "Uso: Vtext nomefile",0 );
  if ( ( infile = fopen( argv[1],"r") ) == NULL )
      gerror( "Non si apre il file di input!",2 );
  printf("\nScansione del file
\033[1m%s\033[0m\n\n",argv[1]);
  while ( getword( item, MAXLENW, infile ) !=
NULL )

      if ( strlen( item ) >= MINLENW && checkw(
item ) == FALSE ) {
          first = insert( first, item );
          ++countword;
      }
  displyalbero( first );
  printf("\n\nParole = %ld ** !ASCII = %d ** ()
= %d ** [] = %d ** {} = %d\n\n",\
  countword,nasc,par1,par2,par3);
}
```

Funzioni di gestione memoria di livello 2

Ecco le funzioni di allocazione di livello 2, come definite secondo lo standard Lattice. I prototipi e le definizioni di queste funzioni sono nel file "stdlib.h".

```
void *getmem( unsigned sbytes);
void *getml( long lbytes );
```

Alloca un blocco di memoria di "sbytes" o "lbytes" bytes nel pool di memoria libera, restituendo il puntatore all'inizio dell'area riservata.

```
int rlsmem( char *p, short sbytes );
int rlsml( char *p, long lbytes );
```

Rilasciano un blocco di memoria precedentemente allocato da getmem(), di cui si fornisce il puntatore "p" e la lunghezza "sbytes" o "lbytes". Restituisce -1 in caso di errore, 0 in caso di operazione riuscita.

Figura 2

blocco e restituendo al sistema (come con free()) il precedente blocco di 500 byte.

Funzioni di livello 2 e 1

Le funzioni di livello 3 sono così chiamate perchè utilizzano funzioni di livello inferiore, 2 e 1, per svolgere il proprio lavoro. Esse consumano un certo spazio supplementare per ogni blocco, dove memorizzano dati che ne consentono una successiva eliminazione automatica in uscita.

Le funzioni di livello più basso, 2 e 1, consentono di sfruttare più efficiente-

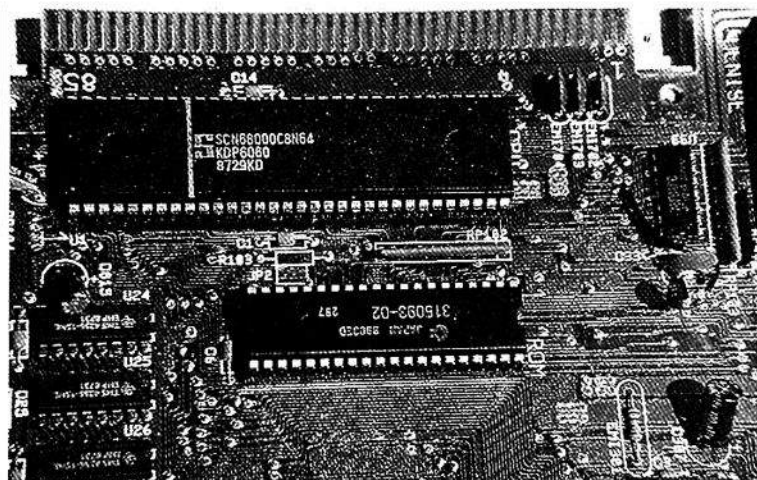
mente la memoria, al prezzo di una minore "assistenza" al programmatore.

Le funzioni di livello due sono riportate in figura 2. Non essendo ancora bene standardizzate nei compilatori di Amiga, dobbiamo per il momento rimandare alla trattazione svolta sui manuali originali. In seguito pubblicheremmo qui, o su AmiGazzetta, listati esemplificativi del loro uso sotto Lattice C ed Aztec C.

Il programma VText.c

Il programma riportato esemplifica l'uso di parecchi concetti esposti sinora: allocazione di memoria, funzioni, costanti, strutture di dati, trasferimneto di parametri e gestione di argomenti e file. Crediamo che commentarlo passo per passo consenta ai lettori di apprendere più rapidamente molti concetti che, esposti teoricamente, sembrerebbero astrusi e difficili da capire.

Lo scopo del programma è scandire un file contenente un testo Ascii, cercando di rilevare le parole scritte male in base ad un banale algoritmo di "parola errata", o meglio "sospetta", implementata come una funzione personalizzabile.

In pratica, il programma legge il file parola per parola memorizzando ciascuna di esse, eliminando quelle parole che sono ripetute almeno due volte (un errore di battitura è difficilissimo che sia ripetuto due volte) e visualizzando soltanto quelle che non soddisfano alcune caratteristiche comuni alle parole scritte correttamente. Ad esempio, una parola con più di due vocali o tre consonanti di fila è giudicata sospetta, quindi visualizzata alla fine.

L'utilità del programma è ovviamente più didattica che pratica; noi lo usiamo di frequente per analizzare i nostri testi (ci vuole poco tempo), scoprendo spesso parole effettivamente scritte male, nonchè per verificare che non vi siano caratteri non Ascii (che provocano problemi ai nostri impaginatori) o parentesi sbilanciate (nel caso vi siano listati nel testo è particolarmente utile).

Commento al listato

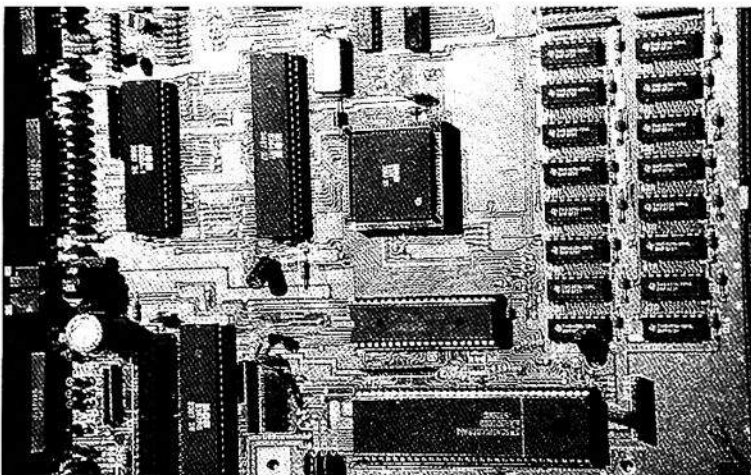
All'inizio del listato troviamo le necessarie inclusioni: **string.h** (per le funzioni di gestione stringa), **stdlib.h** (funzioni di gestione della memoria), **ctype.h** (macro di verifica e conversione caratteri), **stdio.h** (funzioni e strutture di I/O).

Se si usa Lattice C, viene incluso il file con i prototipi delle funzioni DOS, che consente al compilatore di verificare più rigidamente la sintassi delle funzioni usate e di generare codice "in linea", ovvero direttamente del linguaggio macchina per svolgere i compiti delle funzioni di gestione stringa invece che delle chiamate a funzioni di libreria standard (inserite dal linker, ritrovate nel file di libreria **lc.lib**).

Se si usa Aztec C, viene incluso il file gemello **functions.h**, che consente la verifica dei prototipi e la generazione di codice in linea.

Vengono poi definite alcune costanti: la massima lunghezza consentita per ogni parola da memorizzare (**MAXLENW**), la minima lunghezza perchè una stringa sia considerata parola (**MINLENW**) e alcuni valori assegnati per identificare una vocale od una consonante.

E' anche presente una macrodefinizione **isvoc()**, che implementa una funzione che restituisce il valore logico TRUE se la lettera passata come parametro (**c**) è una vocale, il valore FALSE se non lo è.

Segue una serie di variabili globali, "viste" da tutte le funzioni del programma, che conservano parametri utili alle

statistiche finali od al funzionamento del programma: il numero di voci stampate, il numero di caratteri non Ascii, il numero di parentesi tonde, quadre e graffe. Queste ultime tre variabili vengono incrementate ordinatamente quando il programma incontra, nel file Ascii, una parentesi appropriata aperta, e decrementate quando trovano una parentesi chiusa. In questo modo, alla fine dell'esecuzione la stampa dei loro contenuti indica all'operatore quante parentesi "sbilanciate" ci sono, semplicemente in base ad un eventuale valore differente da zero.

Segue la definizione di struttura "nodo", ovvero del gruppo di variabili raggruppate per comodità sotto uno stesso nome. Ciascun elemento "nodo" comprende una variabile che contiene l'indirizzo della parola a cui si riferisce, una variabile che conta quante volte la parola è stata trovata e due puntatori all'elemento (nodo) successivo e precedente. Il programma, in pratica, funziona in base ad un albero di ricerca binario, concetto che amplieremo in una prossima chiacchierata teorica.

La funzione di gestione errori **gerror()** consente di terminare correttamente, da qualunque punto del programma, l'esecuzione generando un messaggio di errore passato come primo parametro.

E' di tipo **void** perchè non restituisce alcun valore alla funzione chiamante, ma termina l'esecuzione con **exit()**.

La funzione **palloc()** è una banale funzione di allocazione di memoria malloc(), trattata precedentemente in forma teorica, che riserva la quantità di memoria necessaria ad un nodo usando l'operatore **sizeof()**. Restituisce NULL, ovvero il valore restituito da malloc(), nel caso in cui l'allocazione non funzioni.

La funzione **savestr()** alloca memoria tramite un malloc() e memorizza una parola nuova. Dopo, usa la funzione **strcpy()** per trasferire la stringa ricevuta come parametro dal programma principale (**main()**) nella memoria allocata e restituisce al chiamante l'indirizzo di memorizzazione della stringa stessa, che sarà ovviamente conservato nell'apposito campo di indirizzo di una struttura nodo, ovvero in **nodo->parola**, come si vede nella funzione **insert()**.

La funzione **displyalbero()** arriva quasi intatta dalla bibbia dei programmatori C: *"The C Programming Language"* di Kernighan e Ritchie (Addison Wesley). Dato un elemento conservato in un albero, ne stampa tutti gli elementi tramite la funzione **printf()**.

La funzione **insert()** inserisce un nuovo elemento nell'albero ed è il fulcro del sistema di gestione dati del programma. Dalla sua struttura si può capire come il programma memorizza i dati da elaborare, ovvero le parole che arrivano dal file Ascii analizzato. In particolare, l'albero è ordinato alfabeticamente, per minimizzare i tempi di ricerca durante la scansione del file.

La funzione **getword()** è di servizio alla principale e riutilizzabile in altri programmi facilmente. Legge una stringa Ascii, forma una parola nel buffer puntato dalla variabile ***s** e aggiorna eventualmente i contatori globali di parentesi.

La funzione **checkw()** prende un puntatore ad una parola e restituisce un valore booleano ("vero" o "falso") corrispondente alla presunta "correttezza" o meno della parola, secondo i semplici criteri prima detti. Un programma più serio userebbe definire una funzione esternamente identica, ma basata internamente sulla scansione di un file di vocabolario, che ricerchi cioè in un file eventuali ricorrenze di ogni parola e restituisca il valore corrispondente ad "errato" solo se la parola è sconosciuta nel vocabolario italiano. Questa procedura è usata da molti programmi commerciali di videoscrittura ed è certamente più efficace, anche se più "mastodontica" e, ovviamente, lenta.

Infine la funzione principale, che svolge semplicemente da interfaccia tra le funzioni descritte e lo Shell Amiga, per mezzo del quale l'utente richiama il programma. La sintassi di chiamata del programma è come segue:

```
VTEXT Nomefile
```

dove **Nomefile** è la definizione standard AmigaDOS di un nome di file. Ad esempio...

```
VTEXT df0:Avv/Cristina
```

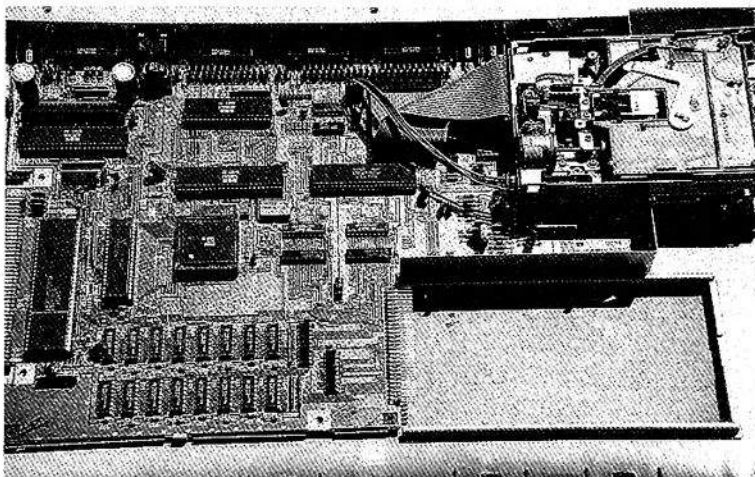
...eseguirà la scansione del file chiamato *Cristina* assunto presente nella directory *Avv* del disco inserito nel drive intero **DF0:**.

Il programma può essere compilato con delle opzioni comuni, ovvero in SAS//Lattice C:

LC -O -vbr -cusfi -Lntv VText.c

Il listato del programma è anche presente sul disco AmiGazzetta numero 10.

Conclusioni

Questa volta siamo stati forse complicati, ma vorremmo con le nostre chiacchierate guadagnare l'attenzione di una larga fascia dei lettori di CCC: dai principianti a chi ha mosso già qualche passo da solo, magari con un buon libro, oppure a scuola con i professori di informatica.

GUIDA MERCATO

Amiga 3000 /40 - L. 5.200.000
Microprocessore Motorola MC68030 - Amiga 3000 a 16 Mhz - Floppy Disk Drive 3 1/2" da 880 Kb - Hard Disk da 40 Mb

A3025/40 - L. 6.200.000
Stessa configurazione del A3016/40 ma con clock a 25 Mhz

A3025/100 - L. 7.050.000
Stessa configurazione del A3016/40 ma con clock a 25 Mhz e Hard Disk da 100 Mb

Amiga 2000 - L. 1.484.000
Microprocessore Motorola MC68000 - Clock 7.16 Mhz - 1 Mbyte RAM - 1 Floppy Disk Drive da 3 1/2", 880 Kbyte

Amiga 500 - L. 786.000
Microprocessore Motorola MC68000 - Clock 7.16 Mhz - 512 Kbyte RAM - 1 Floppy Disk Driver da 3 1/2", 880 Kbyte

A500 FUNLAB - L. 1.260.000
Kit Amiga 500 + tastiera KAWAI + software STEINBERG

C 64

CPU C64C - L. 256.000
Nuovo Personal Computer C64 - 64 Kb RAM - Vastissima biblioteca software disponibile - Chip custom specializzatPio per audio e video - rta seriale Commodore - Porta registratore per cassette - Porta parallela programmabile

C64 GS - L. 180.000
ConfigurazNiuoone C64 Game System

KIT SCUOLA - L. 334.000
Nuovo C64C Scuola

SKATE - L. 347.000
Kit Skateboard

Drive 1541 Kit - L. 280.000
Floppy disk drive 5 1/4" singola faccia - Capacita' 170 Kb Formattati - Compatibile con C64, C128, C128D

1530 - L. 44.000
Registratore a cassette per C64, C128, C128D - Utilizza normali cassette audio per la memorizzazione di programmi e dati

128M - L. 39.000
Mouse 1350 per C64, C128, C128D - Adatto per applicazioni grafiche e per l'ambiente GEOS

STAMPANTI

MPS 1230 - L. 291.000
Stampante a matrice di punti - 9 aghi - 80 colonne - Compatibile con tutti i prodotti Commodore

MPS 1270 - L. 444.000
Stampante a getto d'inchiostro - 80 colonne - Interfaccia Parallela Centronics

MPS 1550C - L. 363.000
Stampante a colori - 9 aghi - 80 colonne di stampa - Compatibile con tutti i prodotti Commodore

MPS 1224C - L. 950.000
Stampante a matrice di punti ad alta qualita' - 24 aghi - 132 colonne - Interfaccia parallela e seriale

MS-DOS

PC 10 10 - L. 709.000
Microprocessore Intel 8088 - Clock 4.77/9.54 Mhz - 640 Kb RAM - 1 Drive 5 1/4" da 360 Kb

PC 10 3 - L. 848.000
Stessa configurazione del PC 10-10 ma con due drive 5 1/4" da 360 Kb

PC 10 FD1 - L. 709.000
Stessa configurazione del PC 10-10 ma con un drive 3 1/2" da 720 Kb

PC 10 FD2 - L. 848.000
Stessa configurazione del PC 10-10 ma con due drive 3 1/2" da 720 Kb

PC 20 3 - L. 1.195.000
Stessa configurazione del PC 10-10 ma con Hard Disk da 20 Mb

PC 20 3 3-2 - L. 1.195.000
Stessa configurazione del PC 10-10 ma con Hard disk da 20 Mb e drive 3 1/2" 720 Kb

PC 30 3 - L. 1.655.000
Microprocessore Intel 80286 - Clock 6/10 Mhz - 1 drive 3 1/2" da 1.44 Mb - Hard Disk 20 Mb - Case Baby

PC 30V 3-2 - L. 1.795.000
Stessa configurazione del PC 30-3 con, in aggiunta, la Super VGA

PC 40 3 1-0 - L. 2.350.000
Microprocessore Intel 80286 - Clock 6/10 Mhz - 1 drive 3 1/2" da 1.44 Mb - Hard Disk 20 Mb

PC 40/40 3 - L. 2.750.000
Stessa configurazione del PC 40-3 1-0 ma con Hard Disk da 40 Mb

PC 50-2 3-0 - L. 3.650.000
Microprocessore a 32 bit Intel 80386sx - Clock a 16 Mhz - 1 Drive 3 1/2" da 1.44 Mb

PC 50-2 3-4 - L. 3.984.000
Come il PC 50-2 3-0 ma con HD 40 Mb

PC 50-2 3-10 - L. 4.750.000
Stessa configurazione del PC 50-2 3-0 ma con Hard Disk da 100 Mb

PC 60-3 5-0 - L. 7.500.000
Microprocessore a 32 bit Intel 80386DX Configurazione base comprendente un Drive 5 1/4" da 1.2 Mb e uno 3 1/2" da 1.44 Mb

PC 60-3 5-8 - L. 7.950.000
Stessa configurazione del PC 60-3 5-0 ma con Hard Disk da 80 Mb

PC 60-3 5-20 - L. 10.200.000
Stessa configurazione del PC 60-3 5-0 ma con Hard Disk da 200 Mb

C286-LT - L. 3.950.000
Notebook con processore Intel 80C286 Configurazione con Drive 3 1/2" da 1.44 Mb
Dotato di Hard Disk da 20 Mb

ACCESSORI

Accessori per MS-DOS compatibili

1352 - L. 75.000
Mouse Microsoft Bus Mouse compatibile - Collegabile direttamente alla serie PC

PC910 - L. 167.000
Floppy Disk Drive interno aggiuntivo da 3 1/2" per la serie PC - Capacita' 720 Kb

PC915 - L. 193.000
Come PC910 ma con capacita' da 1.44 Mb

PC920 - L. 167.000
Floppy Disk Drive interno aggiuntivo da 5 1/4" per la serie PC - Capacita' 360 Kb

PC925 - L. 193.000
Come PC920 ma con capacita' da 1.2 Mb

980 - L. 1.500.000
Hard Disk 80 Mb

9200 - L. 3.500.000
Hard Disk 200 Mb

PC970 - L. 625.000
Streaming Tape 40.6 Mb

PC971 - L. 1.200.000
Streaming Tape 120 Mb

MEM 1/40 - L. 750.000
Espansione 1 Mb per PC 40-3

MEM 2/50 - L. 850.000
Espansione 2 Mb per PC 50-2

PC968-4 - L. 1.850.000
Espansione di memroia da 4 Megabyte compatibile con PC 60-3

PC968-8 - L. 3.100.000
Espansione di memoria da 8 Megabyte compatibile con PC 60-3

Monitor 1402 - L. 125.000
Monitor a fosfori "bianco-carta" - Tubo 12" antiriflesso - Ingresso TTL - Compatibile con tutta la gamma PC

Monitor 1403 - L. 260.000
Monitor monocromatico in standard VGA

Monitor 1404 - L. 168.000
Monitor 14" a fosfori ambra - Compatibile con tutta la gamma PC

Monitor a colori 1930 - L. 880.000
Monitor ad alta risoluzione in standard VGA

Monitor a colori 1950 - L. 1.000.000
Monitor 14" BI-SYNC alta risoluzione - Compatibile con MDA, Hercules, CGA, EGA e VGA

Monitor a colori 1951 - L. 970.000
Monitor a colori VGA (Hitachi)

Accessori per Amiga

A2000TAST - L. 190.000
Tastiera per Amiga 2000

A2000DTV - L. 2.988.000
Desk Top Video per Amiga 2000

A2000DTM - L. 2.792.000
Desk Top Music per Amiga 2000

Drive A1011 - L. 171.000
Drive esterno da 3 1/2" - Capacità 880 Kbyte

Driver A2010 - L. 171.000
Drive interno aggiuntivo da 3 1/2" - Capacità 880 Kbyte - Collegabile ad Amiga 2000

Hard Disk A590 - L. 760.000
Scheda controller - Hard Disk da 3 1/2" 20 Mb - 2 Mb "fast" RAM - Collegabile all'Amiga 500

Scheda Janus A2088 - L. 674.000
Scheda Janus XT - Scheda Bridgeboard per compatibilità MS-Dos (AT) in Amiga 2000 - Clock 8 Mhz - 1 Mb RAM - Floppy Disk Controller - Floppy Disk Drive disegnato per l'installazione all'interno dell'Amiga 2000

Scheda Janus A2286 - L. 1.378.000
Stessa configurazione della scheda A2088 ma con microprocessore Intel 80286

A2630 - L. 3.020.000
Scheda Processore alternativo per 68030 e Unix - Microprocessore Motorola MC68030 - 2 Mb RAM aggiuntivi

A2630/4 - L. 4.074.000
Stessa configurazione della scheda A2630 ma con 4 Mb di RAM aggiuntivi

A2032 - L. 127.000
Modulatore PAL per Amiga

A2320 - L. 460.000
De-Interlacer - Flicker Fixer per Amiga

A2091 - L. 288.000
Controller A2091 per Amiga 2000

A2091/40 - L. 1.194.000
Controller A2091 + Hard Disk da 40 Mb per Amiga 2000

A2058 - L. 798.000
Scheda di espansione per Amiga 2000 fornita con 2 Mb "fast" RAM, espandibile a 4 o 8 Mb

A2300 - L. 298.000
Scheda Genlock semiprofessionale per Amiga 2000

A501 - L. 171.000
Cartuccia di espansione di memoria da 512 Kb per Amiga 500

A520 - L. 42.000
Modulatore RF esterno per Amiga 500 - Permette di connettere qualsiasi televisore all'Amiga 500

Monitor a colori 1084 - L. 472.000
Monitor a colori ad alta risoluzione - Tubo 14" antiriflesso - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor A2024 - L. 1.055.000
Monitor Monocromatico a fosfori "bianco-carta" - Tubo 14" antiriflesso

A2060 - L. 450.000
Scheda ARCNET per Amiga

A2065 - L. 590.000
Scheda ETHERNET per Amiga

Altoparlanti A10 - L. 60.000
Set di amplificazione per l'uscita audio di Amiga

I COMMODORE POINT

LOMBARDIA

Milano
• AL RISPARMIO - V.LE MONZA 204
• BCS - VIA MONTEGANI 11
• BRAHA A. - VIA PIER CAPPONI 5
• E.D.S. - C.SO PORTA TICINESE 4
• FAREF - VIA A. VOLTA 21
• FLOPPERIA - V.LE MONTENERO 31
• GBC - VIA CANTONI 7 - VIA PETRELLA 6
• GIGLIONI - V.LE LUIGI STURZO 45
• L'UFFICIO 2000 - VIA RIPAMONTI 213
• LOGITEK - VIA GOLGI 60
• LU - MEN - VIA SANTA MONICA 3
• MARCUCCI - VIA F.LLI BRONZETTI 37
• MELCHIONI - VIA P. COLLETTA 37
• MESSAGGERIE MUSICALI - GALLERIA DEL CORSO 2
• NEWEL - VIA MAC MAHON 75
• PANCOMMERZ ITALIA - VIA PADOVA 1
• SUPERGAMES - VIA VITRUVIO 38
• 68000 E DINTORNI - VIA WASHINGTON 91
Provincia di Milano
• GINO FERRARI CENTRO HI-FI - VIA MADRE CABRINI 44 - S. ANG. LODIGIANO
• F.LLI GALIMBERTI - VIA NAZIONALE DEI GIOVI 28/36 - BARLASSINA
• TECNOLUX - VIA PIETRO NENNI 5 - BERNATE TICINO
• OGGIONI & C. - VIA DANTE CESANA 27 - CARATE BRIANZA
• AL RISPARMIO - VIA U. GIORDANO 57 - CINISELLO BALSAMO
• GBC - V.LE MATTEOTTI 66 - CINISELLO BALSAMO
• CASA DELLA MUSICA - VIA INDIPENDENZA 21 - COLOGNO MONZESE
• PENATI - VIA VERDI 28/30 - CORBETTA
• EPM SYSTEM - V.LE ITALIA 12 - CORSICO
• P.G. OSTELLARI - VIA MILANO 300 - DESIO
• CENTRO COMPUTER PANDOLFI - VIA CORRIDONI 18 - LEGNANO
• COMPUTEAM - VIA VECELLIO 41 - LISSONE
• M.B.M. - C.SO ROMA 112 - LODI
• L'AMICO DEL COMPUTER - VIA CASTELLINI 27 - MELEGNANO
• BIT 84 - VIA ITALIA 4 - MONZA
• IL CURSORE - VIA CAMPO DEI FIORI 35 - NOVATE MIL.
• I.C.O. - VIA DEI TIGLI 14 - OPERA
• R & C ELGRA - VIA SAN MARTINO 13 - PALAZZOLO MIL.
• ESSEGIEMME SISTEMI SAS - VIA DE AMICIS 24 - RHO
• TECNO - CENTRO - VIA BARACCA 2 - SEREGNO
• NIWA HARD&SOFT - VIA B. BUOZZI 94 - SESTO SAN GIOV.
• COMPUTER SHOP - VIA CONFALONIERI 35 - VILLASANTA
• ACTE - VIA B. CREMIGNANI 13 - VIMERCATE
• IL COMPUTER SERVICE SHOP - VIA PADANA SUPERIORE 197 - VIMODRONE
Bergamo
• D.R.B. - VIA BORGO PALAZZO 65
• TINTORI ENRICO &C. - VIA BROSETA 1
• VIDEO IMMAGINE - VIA CARDUCCI c/o CITTA' DI MERCATO
Provincia di Bergamo
BERTULEZZI GIOVANNI - VIA FANTONI 48 - ALZANO LOMBARDO
• COMPUTER SHOP - VIA VITTORIO VENETO 9 - CAPRIATE SAN GERVASIO
• B M R - VIA BUTTARO 4/T - DALMINE
• MEGABYTE 2 - VIA ROMA 61/A - GRUMELLO
• OTTICO OPTOMETRISTA ROVETTA - P.ZZA GARIBALDI 6 - LOVERE
• COMPUTER POINT - VIA LANTIERI 52 - SARNICO
• A.B. INFORMATICA - STRADA STATALE CREMASCA 66 - URGNANO
Brescia
• MASTER INFORMATICA - VIA F.LLI UGONI 10/B
PROVINCIA DI BRESCIA
• MISTER BIT - VIA MAZZINI 70 - BRENO
• CAVALLI PIETRO - VIA 10 GIORNATE 14 BIS - CASTREZZATO
• VIETTI GIUSEPPE - VIA MILANO 1/B - CHIARI
• MEGABYTE - P.ZZA MALUEZZI 14 - DESENZANO DEL GARDA
• BARESI RINO &C. - VIA XX SETTEMBRE 7 - GHEDI
• INFO CAM - VIA PROVINCIALE 3 - GRATACASOLO
• "PAC-LAND" di GARDONI - CENTRO COM.le - LA CASA DI MARGHERITA D'ESTE - VIA GIORGIONI 21
Como
• IL COMPUTER - VIA INDIPENDENZA 90
• 2M ELETTRONICA - VIA SACCO 3
Provincia di Como
• ELTRON - VIA IV NOVEMBRE 1 - BARZANO'
• DATA FOUND - VIA A. VOLTA 4 - ERBA
• CIMA ELETTRONICA - VIA L. DA VINCI 7 - LECCO
• FUMAGALLI - VIA CAIROLI 48 - LECCO
• RIGHI ELETTRONICA - VIA G. LEOPARDI 26 - OLGIATE COMASCO
Cremona
• MONDO COMPUTER - VIA GIUSEPPINA 11/B
• PRISMA - VIA BUOSO DA DOVARA 8
• TELCO - P.ZZA MARCONI 2/A
Provincia di Cremona
• ELCOM - VIA IV NOVEMBRE 56/58 - CREMA
• EUROELETTRONICA - VIA XX SETTEMBRE 92/A - CREMA
Mantova
• COMPUTER CANOSSA - GAL. FERRI 7
• 32 BIT - VIA C. BATTISTI 14
• ELET. di BASSO - V.LE RISORGIMENTO 69
Provincia di Mantova
• CLICK - ON COMPUTER - S.S. GOITESE 168 - GOITO
Pavia
• POLIWARE - C.SO C. ALBERTO 76
• SENNA GIANFRANCO - VIA CALCHI 5
Provincia di Pavia
• A. FERRARI - C.SO CAVOUR 57 - MORTARA
• LOGICA MAINT - V.LE M.TE GRAPPA 32 - VIGEVANO
• M. VISENTIN - C.SO V. EMANUELE 76 - VIGEVANO
Sondrio
• CIPOLLA MAURO - VIA TREMOGGE 25
Provincia di Sondrio
• FOTONOVA - VIA VALERIANA 1 - S.PIETRO DI BERBENNO
Varese
• ELLE - EFFE - VIA GOLDONI 35
• IL C.TRO ELET. - VIA MORAZZONE 2
• SUPERGAMES - VIA CARROBBIO 13
Provincia di Varese
• BUSTO BIT - VIA GAVINANA 17-BUSTO A.
• MASTER PIX - VIA S.MICHELE 3 - BUSTO A.
• PUNTO UFFICIO - VIA R.SANZIO 8 - GALLARATE
• GRANDI MAGAZZINI BOSSI - VIA CLERICI 196 - GERENZANO
• J.A.C. - C.so MATTEOTTI 38 - SESTO C.

PIEMONTE

Alessandria
• BIT MICRO - VIA MAZZINI 102
• SERV. INFOR. - VIA ALESSANDRO III 47
Provincia di Alessandria
• SONY ITALIANA - VIA G. MANARA 7 - CASALE MONFERRATO
• SGE ELETTRONICA - VIA BANDELLO 19 - TORTONA
• COMPUTER TEMPLE - VIA F. CAVALLOTTI 13 - VALENZA
Asti
• ASTI GAMES - C.SO ALFIERI 26
• RECORD - C.SO ALFIERI 166/3 (Galleria Argenta)
Cuneo
• ROSSI COMPUTERS - C.SO NIZZA 42
Provincia di Cuneo
• PUNTO BIT - C.SO LANGHE 26/C - ALBA
• BOSETTI - VIA ROMA 149 - FOSSANO
• COMPUTERLAND - VIA MAZZINI 30/32 - SALUZZO
Novara
• PROGRAMMA 3 - V.LE BUONARROTI 8
• PUNTO VIDEO - C.so RISORGIMENTO 39/B
Provincia di Novara
• COMPUTER - VIA MONTE ZEDA 4 - ARONA
• ALL COMPUTER - C.SO GARIBALDI 106 - BORGOMANERO
• S.P.A. - C.SO DISSEGNA 21/BIS - DOMODOSSOLA
• ELLIOTT COMPUTER SHOP - VIA DON MINZONI 32 - INTRA
• TRISCONI VALERIA - VIA MAZZINI 90 - OMEGNA
Torino
• ABA ELETTRONICA - VIA C. FOSSATI 5/P
• ALEX COMPUTER E GIOCHI - C.SO FRANCIA 333/4
• COMPUTER HOME - VIA SAN DONATO 46/D
• COMPUTING NEW - VIA M. POLO 40/E
• C.D.M. ELETTR. - VIA MAROCHETTI 17
• DE BUG - C.SO V. EMANUELE II 22
• DESME UNIVERSAL - VIA S.SECONDO 95
• FDS ALTERIO - VIA BORGARO 86/D
• IL COMPUTER - VIA N. FABRIZI 126
• MICRONTEL - C.SO D. degli ABRUZZI 28
• PLAY GAMES SHOP - VIA C. ALBERTO 39/E
• RADIO TV MIRAFIORI - C.SO UNIONE SOVIETICA 381
• SMT ELETTRONICA - VIA BIBIANA 83/bis
Provincia di Torino
• PAUL E CHICO VIDEOSOUND - VIA V.EMANUELE 52 - CHIERI
• BIT INFORMATICA - VIA V. EMANUELE 154 - CIRIE'
• HI - FI CLUB - C.SO FRANCIA 92C - COLLEGNO
• MISTER PERSONAL - VIA CATTANEO 52 - FAVRIA
• I.C.S. - VIA TORINO 73 - IVREA
• DAG - VIA I MAGGIO 40 - LUSERNA S. GIOVANNI
• EUREX - C.SO INDIPENDENZA 5 - RIVAROLO CANAVESE
• DIAM INFORMATICA - C.SO FRANCIA 146 bis - RIVOLI
• FULLINFORMATICA - VIA V.VENETO 25 - RIVOLI
• GAMMA COMPUTER - VIA CAVOUR 3A-3B - SET.TORINESE
Vercelli
• ELETTROGAMMA - C.SO BORMIDA 27 ang. V.Montanara
• ELETTRONICA - STRADA TORINO 15
Provincia di Vercelli
• C.S.I. TEOREMA - VIA LOSANA 9 - BIELLA
• SIGEST - VIA BERTODANO 8 - BIELLA
• REMONDINO FRANCO - VIA ROMA 5 - BORGOSESIA
• FOTOSTUDIO TREVISAN - VIA XXV APRILE 24/B - COSSATO
• STUDIO FOTOGRAFICO IMARISIO - P.ZZA M. LIBERTA' 7 - TRINO

VENETO

Belluno
• UP TO DATE - VIA V. VENETO 43
Provincia di Belluno
• GUERRA COMPUTERS - V.LE MAZZINI 10/A - FELTRE
Padova
• BIT SHOP - VIA CAIROLI 11
• COMPUMANIA - VIA T. CAMPOSANPIERO 37
• D.P.R. DE PRATO R. - V.LO LOMBARDO 4
• G.F. MARCATO - VIA MADONNA DELLA SALUTE 51/53
• SARTO COMPUTER - VIA ARMISTIZIO 79
Provincia di Padova
• COMPUTER SERVICE - BORGO TREVISO 150 - CITTADELLA
Treviso
• BIT 2000 - VIA BRANDOLINI D'ADDA 14
• GUERRA EGIDIO & C. - V.LE CAIROLI 95
Provincia di Treviso
• DE MARIN COMPUTERS - VIA MATTEOTTI 142 - CONEGLIANO
• SIDESTREET - VIA SALVO D'ACQUISTO 8 - MONTEBELLUNA
• FALCON ELETTROAUDIOVIDEO - VIA TERRAGGIO 116 - PREGANZIOL
Venezia
• GUERRA EGIDIO & C. - VIA BISSUOLA 20/A - MESTRE
• TELERADIO FUGA - SAN MARCO 3457
Provincia di Venezia
• GUERRA EGIDIO & C. - VIA VIZZOTTO 29 - SAN DONA' DI PIAVE
• REBEL - VIA F. CRISPI 10 - SAN DONA' DI PIAVE
Verona
• CASA DELLA RADIO - VIA CAIROLI 10
• TELESAT - VIA VASCO DE GAMA 8
• Provincia di Verona
• UBER - CP 0363(RAG.SOC. DERTA) - VIA MASCAGNI 31 - CASTE' D'AZZANO
• FERRARIN - VIA DEI MASSARI 10 - LEGNAGO
• COMPUTERS CENTER - VIA CANTORE 26 - VILLAFRANCA
Vicenza
• ELET. BISELLO - V.LE TRIESTE 427/429
• SCALCHI MARKET - VIA C.A. BALBI 139
Provincia di Vicenza
• SCHIAVOTTO - VIA ZANELLA 21 - CAVAZZALE
• GUERRA E. & C. - V.LE DELLE INDUSTRIE - MONTECCHIO MAGGIORE

FRIULI VENEZIA GIULIA

Gorizia
• E.CO. ELETTRONICA - VIA F.LLI COSSAR 23
Trieste
• AVANZO GIACOMO - P.ZZA CAVANA 7
• COMPUTER SHOP - VIA P. RETI 6
• COMPUTIGI - VIA XX SETTEMBRE 51
• CTI - VIA PASCOLI 4
Udine
• MOFERT 2 - VIA LEOPARDI 21
• R.T. SISTEM UDINE - VIA L. DA VINCI 99
Provincia di Udine
• IL PUNTO ELETTRONICO - VIA VENDRAMIN 184 - LATISANA
• IDRENO MATTIUSSI &C. - VIA LICINIANA 58 - MARTIGNACCO

TRENTINO ALTO ADIGE

Bolzano
• COMPUTER POINT - VIA ROMA 82/A
• MATTEUCCI PRESTIGE - VIA MUSEO 54
Provincia di Bolzano
• RADIO MAIR-ELECTRO - VIA CENTRALE 70 - BRUNICO
• ELECTRO RADIO HENDRICH - VIA DELLE CORSE 106 - MERANO
• ERICH KONTSCHIEDER - PORTICI 313 - MERANO
• ELECTRO TAPPEINER - P.ZZA PRINCIPALE 90 - SILANDRO
Trento
• CRONST - VIA G. GALILEI 25
Provincia di Trento

• AL RISPARMIO - C.SO VERONA 138 - ROVERETO

LIGURIA
Genova
• ABM COMPUTER - P.ZZA DE FERRARI 24 rosso
• CAPRIOTTI G. - IA MAMIANI 4r - SAMPIERDARENA
• C.tro ELET. - VIA CHIARAVAGNA 10 R - VIA SESTRI 69R
• COM.le SOTTORIPA - VIA SOTTORIPA 115/117
• FOTOMONDIAL - VIA DEL CAMPO 3-5-9-11-13 r
• LA NASCENTE - VIA SAN LUCA 4/1
• PLAY TIME - VIA GRAMSCI 3/5/7 rosso
• RAPPR-EL - VIA BORGORATTI 23 R
Imperia
• CASTELLINO - VIA BELGRANO 44
Provincia di Imperia
• CENTRO HI-FI VIDEO - VIA DELLA REPUBBLICA 38 -SANREMO
• CASTELLINO - VIA GENOVA 48 - VENTIMIGLIA
La Spezia
• I.L. ELETTRONICA - VIA V. VENETO 123
Provincia di La Spezia
• I.L. ELETTRONICA - VIA AURELIA 299 - FORNOLA DI VEZZANO
Savona
• CASTELLINO - C.SO TARDY E BENECH 101
Provincia di Savona
• CELESIA ENZA - VIA GARIBALDI 144 - LOANO

EMILIA
Bologna
• EUROELETTRICA - VIA RANZANI 13/2
• MINNELLA ALTA FEDELTA' - VIA MAZZINI 146/2
• MORINI & FEDERICI - VIA MARCONI 28/C
• STERLINO - VIA MURRI 73/75
Provincia di Bologna
• S.C. COMPUTERS - VIA E. FERMI 4 - CASTEL SAN PIETRO
• S.P.E. INFORMATICA - VIA DI MEZZO PONENTE 385 - CREVALCORE
• ARCHIMEDE SISTEMI - VIA EMILIA 124 - S. LAZZARO DI SAVENA
Modena
• CO - EL - VIA CESARI 7
• ORSA MAGGIORE - P.ZZA MATTEOTTI 20
• VIDEO VAL WILLY COMPUTERS - VIA CANALETTO 223
Provincia di Modena
• NEW MEDIA SYSTEM - VIA ROMA 281 - SOLIERA
Parma
• BABARELLI G. - VIA B. PARENTE 14/A/B
Provincia di Parma
• PONGOLINI - VIA CAVOUR 32 - FIDENZA
Piacenza
• COMPUTER LINE - VIA G. CARDUCCI 4
• DELTA COMPUTER - VIA M. DELLA RESISTENZA 15/G

TEGGIO EMILIA
• COMPUTERLINE - VIA SAN ROCCO 10/C
• POOL SHOP - VIA EMILIA S. STEFANO 9/C
Provincia di Reggio Emilia
• MACCHIONI - VIA STATALE 467 - CASALGRANDE
ROMAGNA
Ferrara
• BUSINESS POINT - VIA CARLO MAYER 85
Forlì
• COMPUTER VIDEO CENTER - VIA CAMPO DI MARTE 122
Provincia di Forlì
• TOP BIT - VIA VENETO 12 - FORLIMPOPOLI
• COMPUTER HOUSE - V.LE TRIPOLI 193/D - RIMINI
• EASY COMPUTER - VIA LAGOMAGGIO 50 - RIMINI

REPUBBLICA S. MARINO
Ravenna
• COMPUTER HOUSE - VIA TRIESTE 134
Provincia di Ravenna
• ARGNANI - P.ZZA DELLA LIBERTA 5/A - FAENZA
• ELECTRON INFORMATICA - VIA F.LLI CORTESI 17 - LUGO
• P.L.Z. INFORMATICA - P.ZZA SERCOGNANI 6 - FAENZA

TOSCANA
Arezzo
• DELTA SYSTEM - VIA PIAVE 13
Firenze
• ATEMA - VIA BENEDETTO MARCELLO 1a-1b
• ELETTRONICA CENTOSTELLE - VIA CENTO STELLE 5/a-b
• HELP COMPUTER - VIA DEGLI ARTISTI 15-A
• TELEINFORMATICA TOSCANA -VIA BRONZINO 36
Provincia di Firenze
• WAR GAMES - VIA R. SANZIO 126/A - EMPOLI
• NEW EVM COMPUTER - VIA DEGLI INNOCENTI 2 - FIGLINE VALDARNO
• C.tro INFOR. - VIA ZNOJMO 41 - PONTASSIEVE
• COSCI F.LLI - VIA ROMA 26 - PRATO
• BARBAGLI C. ELET. - VIA F. BONI 80 - PRATO
Grosseto
• COMPUTER SERVICE - VIA DELL UNIONE 7
Livorno
• ETA BETA - VIA SAN FRANCESCO 30
• FUTURA 2 - VIA CAMBINI 19
Provincia di Livorno
• PUNTO ROSSO - VIA BARONTINI 28 - PIOMBINO
Provincia di Lucca
• IL COMPUTER - V.LE COLOMBO 216 - LIDO DI CAMAIORE
• SANTI VITTORIO - VIA ROMA 23 - S. ROMANO GARFAGNANA
• TOP GAMES - VIA S. ANDREA 122 - VIAREGGIO
Massa
• EURO COMPUTER - P.ZZA G. BERTAGNINI 4
Carrara
• RADIO LUCONI - VIA ROMA 24/B
Pisa
• ELECTRONIC SERVICE - VIA DELLA VECCHIA TRANVIA 10
• PUCCINI S.- CP 1199 (RAG.SOC. MAREX) - VIA C.CAMMEO 64
• TONY HI-FI - VIA CARDUCCI
Provincia di Pisa
• M.C. INFORMATICA - VIA DEL CHIESINO 4 - PONTEDERA (PI)
Pistoia
• ELECTRONIC SHOP - VIA DEGLI SCALZI 3
Provincia di Pistoia
• ZANNI &C. - C.SO ROMA 45 - MONTECATINI T.

Siena
• R. BROGI - P.ZZA GRAMSCI 28
• VIDEO MOVIE - VIA GARIBALDI 17
Provincia di Siena
• ELETTRONICA di BIFOLCHI - VIA DI GRACCIANO NEL CORSO 111 - MONTEPULCIANO

LAZIO
• CENTRO INF. - D.R.R. srl - TEL. 06-5565672

UMBRIA
Perugia
• MIGLIORATI - VIA S. ERCOLANO 3-10
Provincia di Perugia
• COMPUTER STUDIO'S - VIA IV NOVEMBRE 18/A - BASTIA UMBRA
• WARE - VIA DEI CASCERI 31 - CITTA'DI CASTELLO
Terni
• CGS SOFTWARE HOUSE - VIA DONIZETTI 71/A

BASILICATA
Matera
• G. GAUDIANO ELECTRONICS - VIA ROMA ang. XX SETTEMBRE 1

PUGLIA
Bari
• ARTEL - VIA GUIDO D'ORSO 9
• COMPUTER'S ARTS - V.LE MEUCCI 12/B
• PAULICELLI S. & F. - VIA FANELLI 231/C
Provincia di Bari
• F. FAGGELLA - C.SO GARIBALDI 15 - BARLETTA
• G FAGGELLA - P.ZZA D'ARAGONA 62A - BARLETTA
• LONUZZO G. - VIA NIZZA 21 - CASTELLANA
• TECNOUFF - VIA RICASOLI 54 - MONOPOLI
• TANGORRA N. - C.SO V.EMANUELE 130/B - TRIGGIANO
Brindisi
• MARANGI E NICCOLI - VIA PROV. SAN VITO 165
Provincia di Brindisi
• MILONE G. - VIA S.F. D'ASSISI 219 - FRANCAVILLA FONTANA
Foggia
• BOTTICELLI G. - VIA SAV. POLLICE 2
• E.C.I. COMPUTER - VIA ISONZO 28
• LA TORRE - V.LE MICHELANGELO 185
Provincia di Foggia
• IL DISCOBOLO - VIA T. SOLIS 15 - SAN SEVERO
Lecce
• BIT - VIA 95 REGG.NTO FANTERIA 87/89
Provincia di Lecce
• TECNO UFFICIO - P.ZZA GIOVANNI XXIII 10 - GALLIPOLI
• CEDOK INFORMATICA - VIA UMBERTO I 116 - TRICASE
Taranto
• ELETTROJOLLY C.tro - VIA DE CESARE 13
• TEA - TEC. ELET. AV. - VIA R. ELENA 101

CAMPANIA
Provincia di Avellino
• FLIP FLOP - VIA APPIA 68 - ATRIPALDA
Benevento
• E.CO. INF. - VIA PEPICELLI 21/25
Caserta
• ENTRY POINT - VIA COLOMBO 31
• O.P.C. - VIA G. M. BOSCO 24
Provincia di Caserta
• M.P. COMPUTER - VIA NAPOLI 30 - MADDALONI
• DAMIANO - C.SO V. EMANUELE 23 -ORTA DI ATELLA
• FUSCO B. - VIA NAPOLI 24 - VAIRANO PATERNORA (FRAZ. VAIRANO SCALO)
• LINEA CONTABILE - VIA OSPEDALE 72/76 - SESSA A. (CE)
Napoli
• BABY TOYS - VIA CISTERNA DELL'OLIO 5/BIS
• CASA MUSICALE RUGGIERO - P.ZZA GARIBALDI 74 (INT. STAZ. F.F. S.S.)
• C.tro ELET. CAMPANO - VIA EPOMEO 121
• CI.AN - GALLERIA VANVITELLI 32
• CINE NAPOLI - VIA S. LUCIA 93/95
• DARVIN - CALATA SAN MARCO 26
• GIANCAR 2 - P.ZZA GARIBALDI 37
• ODORINO - L.GO LALA 22 A-B
• R 2 - VIA F. CILEA 285
• SAGMAR - VIA S. LUCIA 140
• TOP VIDEO - TOP COMPUTER - VIA S. ANNA DEI LOMBARDI 12
• VIDEOFOTOMARKET - VIA S. BRIGIDA 19
Provincia di Napoli
• ELECTRONIC DAY - VIA DELLE PUGLIE 17 - CASORIA
• TUFANO - S.S. SANNITICA 87 KM 7 - CASORIA
• SOF SUD - V.LE EUROPA 59 - CASTEL/MARE DI STABIA
• ELETTRONICA 2000 - C.SO DURANTE 40 - FRATTAMAGGIORE
• SPADARO - VIA ROMANI 93 - MADONNA DELL'ARCO
• GATEWAY - VIA NAPOLI 68 - MUGNANO
• VISPINI & DI VUOLO - VIA A.ROSSI 4 - POMPEI
• SPY CASH & CARRY - P.ZZA ARENELLA 6/A - NAPOLI
• NUOVA INFORMATICA SHOP - VIA LIBERTA' 185/191 PORTICI
• BASIC COMPUTER - C.SO GARIBALDI 34 - POZZUOLI
• V.C. - C.SO SECONDIGLIANO 562/B - SECONDIGLIANO
• F. ELETTRONICA - VIA SARNO 102 - STRIANO
• TECNO - VIA V. VENETO 48 - TORRE DEL GRECO
Salerno
• COMPUMARKET - VIA BELVEDERE 35
• COMPUTER MARKET - C.SO VITTORIO EMANUELE 23
Provincia di Salerno
• KING COMPUTER - VIA OLEVANO 56 - BATTIPAGLIA
• DIMER POINT - V.LE AMENDOLA 36 - EBOLI
• IACUZIO F. - VIA MUNICIPIO 14 - MERCATO SAN SEVERINO
• COMPUTER SERVICE - VIA L.DA VINCI 81 - SCAFATI

CALABRIA
Catanzaro
• C. & G. COMPUTER - VIA F. ACRI 28
• PAONE S. & F. - VIA F. ACRI 93/99
Provincia di Catanzaro
• COMPUTER HOUSE - VIA BOLOGNA (L.GO OSPEDALE) - CROTONE
• RIOLO F.LLI - VIA VENEZIA 1/7 - CROTONE
• ING. FUSTO S. - C.SO NICOTERA 99 - LAMEZIA TERME
Cosenza
• MAISON DE L'INFORMATIQUE - VIA PASQUALE ROSSI 34/C
• SIRANGELO COMP. - VIA N. PARISIO 25
Provincia di Cosenza
• HI-FI ALFANO G. - VIA BALDACCHINI 109 - AMANTIA
• ELIGIO ANNICCHIARICO &C. - VIA ROMA 21 - CASTROVILLARI
• ALFA COMPUTER - VIA NAZIONALE 341/A - CORIGLIANO SCALO
REGGIO CALABRIA
• CONTROL SYSTEM - VIA S.F. DA PAOLA 49 D
• SYSTEM HOU. - VIA FIUME ang. PALESTINO 1
Provincia di Reggio Calabria
• COMPUTER SHOP - V.LE MATTEOTTI 36/38 - LOCRI
• PICIEFFE - C.SO F. S. ALESSIO 19 - TAURIANOVA
SICILIA
• CENTRO INF. - ITALSOFT SRL - TEL. 0935-696090